# Technical Documentation

## *RBFS User Guides*

Version 26.2.1.4 : latest, 30 September 2026

# Table of Contents


1. Overview Guides . . . 3
  1.1. RBFS CLI Overview . . . 3
    1.1.1. RBFS Command Line Interface . . . 3
  1.2. BDS Overview . . . 24
    1.2.1. BDS Overview . . . 24
    1.2.2. BDS Operational Commands . . . 25
2. Routing . . . 31
  2.1. RIB . . . 31
    2.1.1. RIB Overview . . . 31
    2.1.2. RIB Operational commands . . . 38
  2.2. Route Leaking . . . 48
    2.2.1. Route Leaking Overview . . . 48
    2.2.2. Route Leaking Configuration . . . 50
    2.2.3. Route Leaking Operational Commands . . . 52
  2.3. BGP . . . 55
    2.3.1. BGP Overview . . . 55
    2.3.2. BGP Configuration . . . 61
    2.3.3. BGP Operational Commands . . . 95
  2.4. BGP FlowSpec . . . 131
    2.4.1. BGP FlowSpec Overview . . . 131
    2.4.2. BGP FlowSpec Configuration . . . 135
    2.4.3. BGP FlowSpec Operational Commands . . . 137
  2.5. BGP RPKI-RTR Route Validation . . . 151
    2.5.1. BGP RPKI-RTR Prefix Validation Overview . . . 151
    2.5.2. BGP Prefix Validation Using RPKI-RTR Configuration . . . 155
    2.5.3. BGP Route Validation with RPKI-RTR Operational Commands . . . 160
  2.6. IS-IS . . . 165
    2.6.1. IS-IS Overview . . . 165
    2.6.2. IS-IS Configuration . . . 166
    2.6.3. IS-IS Operational Commands . . . 188
  2.7. OSPFv2/v3 . . . 218
    2.7.1. OSPF Overview . . . 218
    2.7.2. OSPF Configuration . . . 221
    2.7.3. OSPF Operational Commands . . . 244
  2.8. LDP . . . 270
    2.8.1. LDP Overview . . . 270
    2.8.2. LDP Configuration . . . 273
    2.8.3. LDP Operational Commands . . . 285

2.9. Policy . . . 298
2.9.1. Policy Overview . . . 298
2.9.2. Policy Configuration . . . 317
2.9.3. Policy Operational Commands . . . 347
2.10. Static Routing . . . 352
2.10.1. Static Routing Overview . . . 352
2.10.2. Static Routing Configuration . . . 353
2.10.3. Static Routing Operational Commands . . . 363
3. Layer 2 Services . . . 366
3.1. EVPN-VPWS . . . 366
3.1.1. EVPN-VPWS Overview . . . 366
3.1.2. EVPN VPWS Configuration . . . 368
3.1.3. EVPN VPWS Operational Commands . . . 378
3.2. BGP-signaled L2VPN . . . 391
3.2.1. BGP-signaled L2VPN Overview . . . 391
3.2.2. Configuring BGP-signaled L2VPN . . . 394
3.2.3. BGP L2VPN Operational Commands . . . 401
3.3. LDP-signaled L2VPN . . . 412
3.3.1. LDP-signaled L2VPN Overview . . . 412
3.3.2. Configuring LDP-signaled L2VPN . . . 414
3.3.3. LDP L2VPN Operational Commands . . . 420
3.4. L2X . . . 425
3.4.1. Layer 2 Cross-Connect . . . 425
3.4.2. L2X Configuration . . . 431
3.4.3. L2X Operational Commands . . . 444
4. Layer 3 Services . . . 453
4.1. L3VPN . . . 453
4.1.1. L3VPN Overview . . . 453
4.1.2. L3VPN Configuration . . . 455
4.1.3. L3VPN Operational Commands . . . 467
5. Subscriber Management . . . 481
5.1. Subscriber Management . . . 481
5.1.1. Subscriber Management Overview . . . 482
5.1.2. Subscriber Management Configuration Overview . . . 513
5.1.3. Subscriber Management Operational Commands . . . 590
5.1.4. Supported Standards . . . 628
5.2. RBFS RADIUS Services . . . 630
5.2.1. RADIUS Services Overview . . . 630
5.2.2. RADIUS Service Solution Overview . . . 633
5.2.3. RADIUS Control . . . 634
5.2.4. RADIUS Accounting . . . 661

5.2.5. RADIUS Counters . . . 664
5.2.6. RADIUS Accounting Adjustment Configuration . . . 669
5.2.7. RADIUS Redundancy . . . 670
5.2.8. Test AAA . . . 671
5.2.9. Appendix A - RADIUS Dictionary . . . 675
5.2.10. Appendix B - RADIUS Standard Attributes Supported . . . 678
5.2.11. Appendix C - Access Deployments . . . 692
5.3. RBFS Carrier-Grade Network Address Translation . . . 693
5.3.1. Carrier-Grade Network Address Translation Overview . . . 693
5.3.2. Carrier-Grade NAT Configuration . . . 700
5.3.3. Carrier Grade NAT Operational Commands . . . 714
5.4. RBFS Carrier-Grade Network Address Translation Appliance . . . 720
5.4.1. CGNAT Appliance Overview . . . 720
5.4.2. CGNAT Appliance Configuration . . . 725
5.4.3. CGNAT Appliance Operational Commands . . . 736
5.5. RBFS HTTP Redirect Service . . . 739
5.5.1. RBFS HTTP Redirect Service Overview . . . 739
5.5.2. HTTP Redirect Service Configuration . . . 741
5.5.3. HTTP Redirect Operational Commands . . . 742
5.6. RBFS Subscriber Filters . . . 744
5.6.1. RBFS Subscriber Filters Overview . . . 744
5.6.2. Subscriber Filters Configuration . . . 745
5.6.3. Subscriber Filters Operational Commands . . . 749
5.7. Lawful Interception . . . 750
5.7.1. Lawful Interception Overview . . . 750
5.7.2. ETSI X1 . . . 753
5.7.3. RADIUS Lawful Interception . . . 755
5.7.4. RBFS Operational State API . . . 756
5.8. L2BSA . . . 758
5.8.1. L2BSA Overview . . . 758
5.8.2. L2BSA Configuration . . . 761
5.8.3. L2BSA Operational Commands . . . 762
5.8.4. L2BSA Operational State API . . . 763
5.9. Redundancy . . . 769
5.9.1. Stateful vs Stateless Redundancy for IPoE . . . 770
5.9.2. Licensing . . . 771
5.9.3. Stateless Redundancy for IPoE Subscribers . . . 771
5.9.4. PADO Delay-Based Redundancy for PPPoE Subscribers . . . 773
5.9.5. Stateful Redundancy for IPoE Subscribers . . . 775
5.9.6. Stateless Redundancy for PPPoE Subscribers . . . 800
6. Forwarding . . . 813

6.1. Interfaces . . . 813
6.1.1. Interfaces Overview . . . 813
6.1.2. Interfaces Configuration . . . 832
6.1.3. Interfaces Operational Commands . . . 858
6.2. ARP/ND . . . 884
6.2.1. ARP/ND Overview . . . 885
6.2.2. ARP/ND Configuration . . . 885
6.2.3. ARP/ND Operational Commands . . . 891
6.3. ACLs . . . 895
6.3.1. ACL Overview . . . 895
6.3.2. ACL Configuration . . . 897
6.3.3. ACL Operational Commands . . . 906
6.4. Port Mirroring . . . 913
6.4.1. Port Mirroring Overview . . . 913
6.4.2. Port Mirroring Configuration . . . 914
6.4.3. Port Mirroring Operational Commands . . . 916
6.5. LAG . . . 917
6.5.1. LAG Overview . . . 917
6.5.2. LAG Configuration . . . 921
6.5.3. LAG Operational Commands . . . 928
6.6. HQoS . . . 932
6.6.1. Hierarchical Quality of Service (HQoS) Overview . . . 932
6.6.2. HQoS Configuration . . . 959
6.7. Configuring QoS Profiles . . . 964
6.7.1. L2 ACL Priority Bit Egress Classifier Configuration . . . 974
6.7.2. Class-Policer-Map Configuration . . . 979
6.7.3. HQoS Show Running-Configuration . . . 1005
6.7.4. HQoS Operational Commands . . . 1012
6.8. LLDP . . . 1026
6.8.1. LLDP Overview . . . 1026
6.8.2. LLDP Configuration . . . 1028
6.8.3. LLDP Operational Commands . . . 1030
7. OAM . . . 1041
7.1. CFM . . . 1041
7.1.1. Ethernet Connectivity Fault Management . . . 1041
7.1.2. CFM Configuration . . . 1042
7.1.3. Operational Commands . . . 1048
7.2. BFD . . . 1051
7.2.1. Bidirectional Forwarding Detection Overview . . . 1051
7.2.2. BFD Configuration . . . 1052
7.2.3. BFD Operational Commands . . . 1062

7.3. Ping and Traceroute . . . 1068
7.3.1. Ping and Traceroute Commands . . . 1068
7.3.2. Ping and Traceroute Operational Commands . . . 1070
8. Multicast . . . 1084
8.1. IGMP . . . 1084
8.1.1. IGMP Overview . . . 1084
8.1.2. IGMP Configuration . . . 1085
8.1.3. IGMP Operational Commands . . . 1093
8.2. PIM . . . 1100
8.2.1. PIM Overview . . . 1100
8.2.2. PIM Configuration . . . 1101
8.2.3. PIM Operational Commands . . . 1106
8.3. Multicast VPN . . . 1116
8.3.1. MVPN Overview . . . 1116
8.3.2. MVPN Configuration . . . 1117
8.3.3. MVPN Operational Commands . . . 1123
9. Security . . . 1126
9.1. Securing Management Plane . . . 1126
9.1.1. Securing the Management Plane Overview . . . 1126
9.1.2. RtBrick Token . . . 1128
9.1.3. Role Based Access Control (RBAC) . . . 1133
9.1.4. SSH with TACACS+ . . . 1136
9.2. Securing Control Plane . . . 1143
9.2.1. Securing the Control Plane Overview . . . 1143
9.2.2. Control Plane Security Configuration . . . 1147
9.2.3. Control Plane Security Operational Commands . . . 1154
9.3. Local User Management . . . 1164
9.3.1. Local User Management Overview . . . 1164
9.3.2. Local user management Configuration . . . 1165
9.3.3. LUM Operational Commands . . . 1178
9.4. sFlow . . . 1179
9.4.1. RBFS sFlow Technology . . . 1180
9.4.2. sFlow Configuration . . . 1182
9.4.3. sFlow Operational Commands . . . 1190
10. Telemetry . . . 1193
10.1. TSDB . . . 1193
10.1.1. Observability . . . 1193
10.1.2. Time-series Configurations in RBFS . . . 1200
10.1.3. Observability Operational Commands . . . 1219
10.2. Resmon . . . 1223
10.2.1. Resource Monitoring Overview . . . 1223

10.2.2. Resource Monitoring Configuration . . . 1231
10.2.3. Resource Monitoring Operational Commands . . . 1233
10.3. SNMP . . . 1262
10.3.1. SNMP Overview . . . 1262
10.3.2. SNMP Configuration . . . 1264
11. Infrastructure . . . 1273
11.1. NTP . . . 1273
11.1.1. NTP Overview . . . 1273
11.1.2. NTP Configuration . . . 1274
11.1.3. NTP Operational Commands . . . 1276
11.2. Clock Synchronization . . . 1279
11.2.1. Clock Synchronization . . . 1279
11.2.2. Clock Synchronization Configuration . . . 1281
11.2.3. Clock Synchronization Operational Commands . . . 1286
11.3. Logging . . . 1303
11.3.1. Logging Overview . . . 1303
11.3.2. Logging Configuration . . . 1308
11.3.3. Logging Operational Commands . . . 1320
11.3.4. RBFS Log Reference . . . 1377
11.4. LED Control . . . 1549
11.4.1. LED Overview . . . 1549
11.4.2. LED Definitions . . . 1549
11.5. IPMI . . . 1561
11.5.1. IPMI Overview . . . 1561
11.5.2. IPMI Configuration . . . 1561
11.5.3. IPMI Operational Commands . . . 1563
11.6. Inband Management . . . 1566
11.6.1. In-band Management Overview . . . 1566
11.6.2. In-band Management Configuration . . . 1567
11.6.3. In-Band Management Operational Commands . . . 1577
11.7. DHCP Relay/Proxy . . . 1578
11.7.1. DHCP Relay Overview . . . 1578
11.7.2. DHCP Relay Configuration . . . 1580
11.7.3. DHCP Relay Operational Commands . . . 1583

The RtBrick Full Stack (RBFS) User Guides provide comprehensive documentation covering configuration and operational commands for the protocols and applications in RBFS.

The guides are structurally divided into the following categories:

- **Overview Guides**: Provide operational details and insights into the RBFS Command-Line Interface (CLI) and the Brick Data Store (BDS).
- **Routing User Guides**: Detailed guides for configuring BGP, IS-IS, OSPFv2/v3, Static Routing, LDP, BGP FlowSpec, BGP RPKI, route leaking, and Policy management.
- **Layer 2 Services Guides**: Provide the necessary configuration and operational commands to set up Layer 2 virtual private networks and services on the RBFS platform.
- **Layer 3 Services Guides**: Provide configuration and operational commands for implementing layer 3 networking functionalities within the RBFS platform.
- **OAM (Operations, Administration, and Maintenance) Guides**: Configurations for link monitoring and fault management using CFM, BFD, Ping, and Traceroute.
- **Subscriber Management Guides**: Detailed documentation for deploying RBFS as a Broadband Network Gateway (BNG), managing RADIUS services, Carrier-Grade NAT (CGNAT), HTTP redirect services, Subscriber Filters, Lawful Interception, L2BSA, and subscriber redundancy (IPoE and PPPoE stateless redundancy).
- **Forwarding Guides**: Configure basic physical and logical packet forwarding properties, including Interfaces, ARP/ND, Access Control Lists (ACLs), Port Mirroring, Link Aggregation Groups (LAGs), Hierarchical Quality of Service (HQoS), and LLDP.
- **Multicast Guides**: Procedures for streaming media control via IGMP, PIM, and Multicast VPN (MVPN).
- **Security Guides**: Provide configuration and operational instructions to safeguard and monitor the RBFS environment. They focus on protecting the management and control planes, enforcing local user access controls, and implementing protocol-level defence mechanisms like BGP FlowSpec, BGP RPKI, and sFlow.
- **Telemetry**: Details data collection from various RBFS components to support monitoring tools, including TSDB (Time Series Database), Resmon (Resource Monitor), SNMP, and sFlow streaming.
- **Infrastructure**: Focuses on essential system services such as NTP, clock synchronisation, system logging, LED control, IPMI, In-band Management, and DHCP Relay/Proxy operations.

The following guides are available under the RBFS User Guides section:

- Overview Guides
- Routing
- Layer 2 Services
- Layer 3 Services
- Subscriber Management

- Forwarding
- OAM
- Multicast
- Security
- Telemetry
- Infrastructure

# 1. Overview Guides

**CLI Overview Guide**:

This guide explains how to use the RBFS Command-Line Interface (CLI) to configure, monitor, and troubleshoot your system. You will get familiar with CLI prompts and modes, use helpful shortcuts, view command history, browse the CLI tree, review logs, and manage configuration changes. This guide equips you with the knowledge needed to work with the RBFS CLI.

**BDS Overview**:

An introduction to the underlying architectural brain of RBFS. It explains how BDS operates as a high-performance, in-memory state database using a Pub-Sub model to pass data between system's microservices.

The following guides are available under the Overview Guides section:

- RBFS Command Line Interface
- BDS Overview

## 1.1. RBFS CLI Overview

### 1.1.1. RBFS Command Line Interface

**RBFS CLI Overview**

RBFS command line interface is a primary user interface that enables you to interact with RBFS for monitoring, configuring, debugging, and maintaining the system. RBFS command line interface, that runs on top of the Debian shell, provides a rich set of commands which allow you to execute various operations on the system.

RBFS CLI commands are organized in hierarchies based on their functionalities. Commands, which are used to execute the same type of functions, have the same hierarchy. For example, to display information, you can use commands that start with 'show'. `Delete` command, in RBFS, is used to remove an existing configuration.

The RBFS command-line interface has three modes: Configuration mode, Operation mode, and Debug mode.

**Operational mode:** This is the default mode of RBFS command line interface. Operational mode allows you to execute the operational commands such as show commands to view or monitor various system configuration and its current state.

**Configuration mode:** Configuration mode allows to execute various configurations for services or features. It also allows you to view the information for the existing configurations.

**Debug mode:** It allows you to execute troubleshooting or debugging operations in the RBFS system.

**Using the CLI**

The following are some of the utilities which help you working with the CLI faster and easier.

**Complete Partially Typed Commands**:

You can press `Tab` key to complete a partially typed command. It helps you work with commands faster.

**Command Options and Description**:

If you do not know the options available for a command and the purposes of the options, you can enter the question mark symbol (?). It displays all the available command options and descriptions for that commands.

In any of the modes, if you type the question mark symbol (?), the CLI displays a set of commands which can be executed in that particular mode.

When you execute configurations through CtrlD, and then with the Command Line Interface, it results in error when you commit the configuration through the CLI. The reason is that CtrlD directly interacts with the backend applications (BDS and CONFD) and these changes are not synced with the CLI.

#### Launching the RBFS CLI

The following example shows how to start the RBFS CLI.

```
supervisor@rtbrick>LEAF01:~$ cli
supervisor@rtbrick>LEAF01: op>
```

'op>' shows you are in operational mode.

When you log in to the RBFS CLI, it displays container, image, and platform metadata, and the RBFS license details as the following:

```
supervisor@rtbrick.net:~ $ cli

+-------------------------------------------------------------------+
|                                                                   |
    RBFS container rtbrick running on rtbrick.net:
        Date:         Mon Jul 13 11:45:50 AM UTC 2026
        Uptime:       up 2 hours, 9 minutes

    Image and Platform metadata:
        UUID:          9e74a1ba-fc42-40ee-9a76-b123456789123
        Version:       26.2.1
        Role:          accessleaf
        Format:        lxd
        Build date:    2026-07-10 13:03:09 UTC
```

```
        Based on:      Debian GNU/Linux 12 (bookworm)
        Platform:      q2c
        Model:         s9600-72xc
        Serial Number: XBC1F9V300002
        Firmware:      v3.3.0

    RBFS License:
        Active License Rtbrick
          Start date: Fri Mar 06 00:00:00 GMT +0000 2026
          End date: Sat Mar 06 23:59:00 GMT +0000 2027
          Days Left: 236
|                                                                  |
+------------------------------------------------------------------+
```

#### CLI Prompt

This section introduces you to three types of prompts:

- **RBFS CLI prompt**: The RBFS CLI prompt is the command-line interface (CLI) used to interact with the RtBrick FullStack (RBFS) software. Upon logging into RBFS, you will see the CLI prompt in Operational State mode. To switch to a different mode, see the Switching CLI Modes section.
- **Container shell prompt**: The container shell prompt refers to the command-line interface that appears when you access a Linux container environment where RBFS is installed.
- **Host prompt**: The RtBrick Host prompt refers to the command-line interface that appears when using the RBFS software after it is installed.

#### RBFS CLI Prompt

The RBFS CLI prompt reflects the static hostname and host OS hostname. In RBFS, the static hostname is the container name and the dynamic hostname is derived from DHCP.

The format of the RBFS CLI prompt is as follows:

`<username> @ <static_hostname> > <hostname.host-os>: <mode>`

| Option | Description |
|---|---|
| `<username>` | This specifies the user-defined name. |
| `static_hostname` | This is specified in the `/etc/hostname` directory and always corresponds to the container name. |
| `hostname.host-os` | This is specified in the `/etc/hostname.host-os` directory and reflects a DHCP-learned hostname. |
| `mode` | This identifies the CLI mode, such as `cfg` (Configuration mode), `op` (Operation mode (default mode)), or `dbg` (Debug) mode. To switch between modes, enter the "switch-mode <mode>" command. For details on switching between modes, see the Switching CLI Modes section below. |

#### Switching CLI Modes

RBFS CLI has three modes: Configuration mode, Operation mode, and Debug mode. You can enter `switch-mode` command to change the CLI mode.

The following example shows the options available in the "switch-mode" command.

```
supervisor@rtbrick>LEAF01: op> switch-mode ?
    config                Enter a given mode
    debug                 Enter a given mode
    operation             Enter a given mode
```

The following examples shows how to switch between modes and how the CLI prompt changes with each mode.

**Switching from operation (op) to configuration (cfg) mode:**

```
supervisor@rtbrick>LEAF01: op> switch-mode config
Activating syntax mode : cfg [config]
supervisor@rtbrick>LEAF01: cfg>
```

**Switching from configuration (cfg) to operation (op) mode:**

```
supervisor@rtbrick>LEAF01: cfg> switch-mode operation
Activating syntax mode : op [operation]
supervisor@rtbrick>LEAF01: op>
```

**Switching from configuration (operation) to debug (debug) mode:**

```
supervisor@rtbrick>LEAF01: op> switch-mode debug
Activating syntax mode : dbg [debug]
supervisor@rtbrick>LEAF01: dbg>
```

#### Container Shell Prompt

The container shell prompt reflects the host OS's static hostname and hostname.

The format of the container shell prompt is as follows:

<username> @ <static_hostname> > <hostname.host-os>:<current_directory> $

| Option | Description |
|---|---|
| `<username>` | This specifies the user-defined name. |
| `static_hostname` | This is specified in the `/etc/hostname` directory and always corresponds to the container name. |

| Option | Description |
|---|---|
| `hostname.host-os` | This is specified in the `/etc/hostname.host-os` directory and reflects a DHCP-learned hostname. |
| `<directory>` | This specifies your current working directory. |

Example:

```
rtbuser@rtbrick>LEAF01:~ $ cat /etc/hostname
rtbrick

rtbuser@rtbrick>LEAF01:~ $ cat /etc/hostname.host-os
LEAF01
```

#### RtBrick Host Prompt

You can log in into RtBrick Host using SSH port 1022, which applies to both in-band and out-of-band access.

The format is as follows:

ssh <user>@<host-ip> -p 1022

| Option | Description |
|---|---|
| `<user>` | This is the unique identifier you assign to a user account while installing RBFS. |
| `<host-ip>` | This specifies the host IP address. |

```
ssh rtbuser@192.0.2.0
rtbuser@onl:~ $
```

#### Turning Paging On/Off

To turn the paging on or off, use the following command:

**paging [on | off]**

- off - Pagination will be turned off for the commands that span more than screen length
- on - Pagination will be turned on for the commands that span more than screen length

Example:

```
supervisor@rtbrick>LEAF01: cfg>  paging on
```

#### Turning On or Off CLI Colors

To enable or disable the colors in the CLI output, use the following command:

**color [on | off]**

Example:

```
supervisor@rtbrick>LEAF01: cfg> color on
```

#### Accessing CLI Logs

RBFS supports sending command history log messages to Graylog, a log management software that enables real-time analysis of log messages.

The command history logs help you to understand which user has executed a specific command across multiple CLI sessions.

The log format for CLI command history logs is: *User '%s' executed command '%s'*.

System logging is implemented for RESTCONF.

> For RESTCONF error logs, do not set the log level to 'info'. If you set the log level to `info`, logs are generated for all the `restconfd` requests.

### Basic CLI Features

#### Viewing CLI Tree

The `show cli tree` command displays all RBFS CLI commands and command options that exist.

- `show cli tree`: This command provides you the list of all RBFS CLI commands and options that are available.
- `show cli tree detail`: The detail option for the command displays additional information such as minimum and maximum values allowed.

#### Viewing Command History

The `history` command enables you to view the previously executed commands. You can execute the command in any of the CLI modes.

**history**

Example:

```
supervisor@rtbrick>LEAF01: op> history
show config set
exit
```

```
show config set
load config test.json
load config obj.json
show config set
exit
show config set
load config test.json
switch-mode config
load config test.json
load config obj.json
exit
switch-mode config
show config set
load config test.josn
load config obj
load config obj.json
exit
show config set
load config obj.json
load config test.json
exit
show bd running-status
load config test.json
show config set
exit
show bd running-status
exi
show datastore confd table test index index2
exit
```

#### Setting Configuration Parameters

The `set` command is used to configure parameters.

```
supervisor@rtbrick: cfg> set system ntp server MyNTPServer ipv4-address 10.1.1.100
```

#### Deleting Configuration Parameters

The `delete` command is used to remove existing configuration parameters.

```
supervisor@rtbrick: cfg> delete system ntp server MyNTPServer ipv4-address 10.1.1.100
```

#### Viewing Core Dump Files

When a router crashes, it is useful to obtain a complete copy of the memory image, also known as core dump, to identify the cause of the crash.

The `show core-dump stack-trace` command displays core dump data generated by the system. It

includes information such as Core Dump File Name, Process Name/ID, Time of Core Dump, Core Dump Size, Signal Information, Process State, and so on.

You can use the command in the following ways:

- `show core-dump` lists the core dump files.
- `show core-dump stack-trace` displays all the stack traces (list of the function calls) for all valid core files.
- `show core-dump stack-trace <file-name>` displays the stack traces for a specified file.
- `show core-dump stack-trace <file-name> per-thread` displays stack trace for all threads for the specified file.

> Each process is allowed to retain up to three core files. When a new core file is generated, the oldest file will be automatically deleted. It improves disk space management while retaining recent crash data for debugging.

**clear core-dump**

- Use the `clear core-dump` to clear output populated.

Additionally, you can specify a file name to remove core dump data for that file.

**clear core-dump** <file-name>

#### Viewing Hardware Resource Usage Limit Information

In RBFS, you can view the hardware resource usage limit details.

Run the following command:

`show hardware limits`

Example:

```
supervisor@rtbrick>rtbrick.net: op> show hardware limits
Hardware resources:
  Module: fib
    ASIC                            : q2c
    Role                            : accessleaf
    Model                           : agcva48s
    IPv4 route count                : 1200000
    IPv6 route count                : 250000
  Module: fib
    ASIC                            : q2c
    Role                            : accessleaf
    Model                           : as7946-74xkb
```

```
    IPv4 route count                : 1200000
    IPv6 route count                : 250000
  Module: fib
    ASIC                            : q2c
    Role                            : accessleaf
    Model                           : as7946-30xb
    IPv4 route count                : 1200000
    IPv6 route count                : 250000
  Module: fib
    ASIC                            : q2c
    Role                            : accessleaf
    Model                           : s9600-72xc
    IPv4 route count                : 1200000
    IPv6 route count                : 250000
  Module: bgp
    ASIC                            : q2c
    6PE label                       : 2
  Module: confd
    ASIC                            : q2c
    Max MTU profile                 : 8
    Max L3 MTU profile              : 3
    Max subscriber MTU profile      : 6
    Max physical MTU profile        : 8
  Module: rd
    ASIC                            : q2c
```

#### Rebooting Containers and Hosts

The `reboot` command allows you to restart containers and hosts.

**reboot** <option>

| Option | Description |
|---|---|
| `-` | Without any option, this command allows you to reboot a container (default). You are prompted to confirm rebooting the container when you enter this command. You must answer yes or no. |
| `container` | This command allows you to reboot a container. You are prompted to confirm rebooting the container when you enter this command. You must answer yes or no. |
| `container-and-confirm` | This command reboots the container without prompting yes/no. |
| `device` | This command allows you to reboot a device. You are prompted to confirm rebooting the device when you enter this command. You must answer yes or no. |
| `device-and-confirm` | This command reboots the device without prompting yes/no. |

Example:

```
supervisor@rtbrick>LEAF01: cfg> reboot container
```

#### Viewing System Version Details

To display the version details of RBFS and its various components, use the `show version` command.

**show version**

Example:

```
supervisor@ixr_pe1>srv3.nbg1.rtbrick.net: op> show version
UUID        : 2abb4250-2a14-4e5c-84e2-6785eee158f8
Version     : 22.6.0-g4internal.20220620060710+Bfs0000bgpauthlatest.C3abc099d
Role        : spine
Platform    : virtual
Format      : lxd
Build date  : 2022-06-20 06:07:10 UTC
Component                        Version
Timestamp                  Branch
alertmanager                     0.20.1001-internal.20220613124702+Bdevelopment....
2022-06-07 20:01:29        development
cligen                           0.1.0-internal.20220613140225+Bdevelopment.C9457c97b
2022-06-07 20:00:33        development
clixon                           4.3.1-internal.20220618124913+Bdevelopment.C85593b60
2022-06-13 11:48:32        development
<...>
```

#### Viewing Date and Time

To display system date and time, use the `date` command.

**date**

Example:

```
supervisor@rtbrick>LEAF01: op> date
Thu Apr 28 09:56:32 UTC 2022
```

#### Viewing the Running configuration

To display the running configurations, enter the `show config` command.

Example:

```
supervisor@rtbrick>LEAF01: cfg> show config
```

#### Viewing a Specific Hierarchy

To display the configurations under a specific hierarchy, enter the `show config <hierarchy>` command.

```
supervisor@rtbrick: cfg> show config ?
  access                Global access configuration
  daemon-options        List of daemon options
  global                Global configurations
  instance              Network instance configuration
  <...>
```

#### Viewing Configurations in a Specific Format

The `show config` command displays the current committed configurations of the system. By default, this command displays the configurations in a `json` format.

**show config** <format>

You can also specify the format explicitly, if needed. The available display formats are:

- **json**: Display configurations in JSON format
- **set**: Display configurations in CLI format (similar to commands executed)
- **netconf**: Display configurations in XML format
- **text**: Display configurations in textual format (similar to YANG definition)

The following example shows how configurations are displayed in the text format.

```
supervisor@rtbrick>LEAF01: op> show config text
daemon-options {
    instance-name *;
    afi *;
    safi *;
    bd-type bgp.appd;
    bd-name bgp.appd.1;
}
daemon-options {
    instance-name *;
    afi *;
    safi *;
    bd-type bgp.iod;
    bd-name bgp.iod.1;
}
```

```
interface {
    name lo-0/0/0;
    unit {
        unit-id 0;
        address {
            ipv4 {
                prefix4 198.51.100.75/24;
            }
            ipv6 {
                prefix6 2001:db8:0:110::/32;
            }
        }
    }
}
<...>
```

To view configurations in the set format, use the `show config set` command.

Example:

```
supervisor@rtbrick>LEAF01: cfg>  show config set
set interface ifp-0/0/1
set interface ifp-0/0/1 ifp-id 1
set interface ifp-0/0/2
set interface ifp-0/0/2 ifp-id 2
set instance blue
set instance blue protocol bgp address-family ipv4 multicast
set instance blue protocol bgp address-family ipv6 unicast
set instance red
set instance red protocol bgp address-family ipv4 unicast
set instance red protocol bgp address-family ipv6 unicast
```

#### Viewing Configuration in a Specified Hierarchy

To view configuration in a specified hierarchy, use the following command:

```
supervisor@rtbrick>LEAF01: cfg> show config set instance red protocol bgp
set instance red protocol bgp address-family ipv4 unicast
set instance red protocol bgp address-family ipv6 unicast
```

#### Committing CLI Configurations

To apply the configurations, use the `commit` command.

The following table provides information about the commit command and its extended options, along with their descriptions.

| Command | Description |
|---|---|
| `commit` | Applies the configuration to make it active on the system. |
| `commit and-quit` | Commits the configuration changes and exits configuration mode. |
| `commit <message>` | Commits the configuration and associates a user-defined message with the commit for tracking purposes. Value range: 1 -128. |
| `commit cancel` | Cancels a pending confirmed commit and reverts to the previous active configuration. |
| `commit confirmed` | Commits the configuration temporarily and automatically rolls back unless explicitly confirmed within the default timeout. |
| `commit confirmed` | Commits the configuration with a user-defined rollback timeout (in seconds) unless confirmed. Value range: 1 - 4294967295. |
| `commit confirmed <message>` | Performs a confirmed commit with both a custom timeout and an associated commit message. Value range: 1 - 4294967295. |
| `commit confirmed <message>` | Performs a confirmed commit with a user-defined message using the default rollback timeout. Value range: 1 - 128. |
| `commit switch-to-op` | Commits the configuration and switches the session to operational mode. |

The following example shows how to commit your changes using the `commit` command.

```
supervisor@rtbrick>LEAF01:~$ cli
supervisor@rtbrick>LEAF01: op> switch-mode config
supervisor@rtbrick>LEAF01: cfg> <cli command goes here>
supervisor@rtbrick>LEAF01: cfg> commit
```

When you exit CLI configuration with uncommitted changes, a reminder text appears saying that you have changes to commit, as shown in the following example:

```
supervisor@rtbrick>LEAF01: cfg> exit
Uncommitted changes are present
1. Discard the changes and exit
2. Commit the changes and exit
3. Keep the changes and exit [Default behavior]
Enter one of the above choice to proceed :
```

#### Viewing Uncommitted Changes

To view the uncommitted changes, use the `show diff` command. The `show diff` command displays output in JSON format, whereas the the `show diff set` command displays the output in CLI format.

```
supervisor@rtbrick>LEAF01: cfg> show diff

supervisor@rtbrick>LEAF01: cfg> set interface ifp-0/0/3 ifp-id 3
```

```
supervisor@rtbrick>LEAF01: cfg> set interface ifp-0/0/4 ifp-id 4
supervisor@rtbrick>LEAF01: cfg> show diff set
+set interface ifp-0/0/3
+set interface ifp-0/0/3 ifp-id 3
+set interface ifp-0/0/4
+set interface ifp-0/0/4 ifp-id 4
supervisor@rtbrick>LEAF01: cfg> show diff
 }
+interface {
+    name ifp-0/0/3;
+    ifp-id 3;
+}
+interface {
+    name ifp-0/0/4;
+    ifp-id 4;
+}
 instance {
```

#### Adding a Configuration Description

An in-line description or comment can be added to a system configuration to describe it.

**set system config-description** <description>

Example:

```
supervisor@rtbrick>LEAF01: cfg> set system config-description "This is sample test
configuration"
supervisor@rtbrick>LEAF01: cfg> commit
supervisor@rtbrick>LEAF01: cfg> show config
{
  "ietf-restconf:data": {
    "rtbrick-config:system": {
      "config-description": "This is sample test configuration"
    }
  }
```

#### Saving Configuration

To save configurations, enter the following command:

```
supervisor@rtbrick>LEAF01: cfg> save config my_config.json
```

- Ensure that you use .json at the end of the filename.
- The configuration will be saved to the current working directory of CLI executable.

#### Deleting the Entire Running Configuration

The `discard` command is used to discard configuration changes. To delete the entire running configurations (both committed and uncommitted configurations) at a time, use the `discard all` command.

Example:

```
supervisor@rtbrick>LEAF01: cfg> discard all
```

#### Discarding the Uncommitted Configuration

To discard the uncommitted configuration, enter the following command:

```
supervisor@rtbrick>LEAF01: cfg> discard
```

#### Viewing the Configuration Differences in the Current and Previous Versions

In RBFS, you can view the configuration differences between the current and the previous versions.

**show diff** <number>

```
supervisor@rtbrick>LEAF01: cfg> show diff 2
 system {
-    secure-management-status false;
+    secure-management-status true;
 }
```

#### Rolling Back to a Previously Committed Configuration

To rollback to a specific configuration prior to the most recently committed one, use the following command:

**rollback** <number>

`number`: Specifies the rollback ID. Range: 1 through 49. 0 refers to the active configuration, 1 refers to the most recent previous configuration. Default: 1

For example, to rollback to rollback ID 2, use the following command:

```
supervisor@rtbrick>LEAF01: cfg> rollback 2
```

#### Rolling Back to a Specific Version of Software Configuration

To rollback to a specific version of the software configuration, use the following command:

**rollback commit-id** <commit-hash>

Example:

```
supervisor@rtbrick>LEAF01: cfg> rollback commit-id 29d5db038c1920fdsdsdsdsdsd323232
```

#### Loading Configuration

To load configurations, enter the following command:

**load config** <filename> <option>

The options include `merge` and `replace`. You can specify `merge` after the file name to merge the configuration with the running configuration. Specify `replace` to replace the running configuration with the new one. Without any option, it replaces the running configuration, by default.

```
supervisor@rtbrick>LEAF01: cfg>  load config <filename>
```

- Ensure that you use '.json' at the end of the filename.
- Remember to commit your changes after loading.

#### Watch Mode

The watch mode configuration allows users to enable or disable real-time command monitoring, periodically refresh output (in-place refresh), and highlight changes between updates.

To run commands with watch mode enabled:

1. Set the watch-mode to 'on' and specify the time interval (in seconds) for polling updates.
2. Execute the desired show command.
3. The command output will refresh at the specified interval, highlighting any changes since the last update.

Currently, the watch-mode feature supports the following CLIs that allow users to retrieve various statistics and counter information.

```
show interface statistics
show interface <name> statistics
show hostpath counters
show hostpath shm counters
```

```
show l2x statistics <l2x name>
show trap statistics
show trap statistics <trap type>
show resource statistics
show transceiver
show transceiver <interface>
show memory usage
show sensor fan
show sensor power-supply
show sensor temperature
show bgp peer statistics
show bgp peer statistics instance <instance_name>
show bgp peer statistics instance <instance_name> peer <peer_name>
show bgp peer statistics peer <peer_name>
show isis interface statistics
show ldp statistics
show ldp statistics instance <instance-name>
show ldp statistics instance ldp-id <peer>
show address-translation trap statistics
show ospf statistics interface
show policy statistics <bd>
show policy statistics <bd> policy-name <policy_name>
show policy statistics <bd> app-key <app_key>
show acl statistics /trap/rule/ordinal
show dynamic-service statistics
show datastore statistics
show datastore <bd> object-template statistics
show datastore <bd> array-template statistics
show datastore <bd> bd connection statistics peer <peer>
show commit statistics
show redundancy client statistics
```

Syntax:

**watch-mode** (on | off)
**watch-mode time-interval** <2-120>

| Attribute | Description |
|---|---|
| watch-mode off | Disables real-time command monitoring. This is the default state. |
| watch-mode on | Enables real-time command monitoring. |
| watch-mode time-interval <2-120> | Sets the polling interval in seconds for monitoring command execution. The value must be between 2 and 120 seconds. Default is 2 seconds. |

The following example sets the command execution monitoring to 5 seconds.

```
supervisor@S1-STD-2-2015>LEAF01: cfg> watch-mode time-interval 5
supervisor@S1-STD-2-2015>LEAF01: cfg> watch-mode on
supervisor@S1-STD-2-2015>LEAF01: cfg> show memory usage
```

The result of the `show memory usage` command with watch mode enabled is as follows:

```
Every 5.0s: cli -m cfg -1 show memory usage                                    S1-STD-2-2015: Tue Jan 20 05:41:26 2026

Name     Total          Used           Free           Shared       Buffers      Cached
RAM      125.71 GiB     9.45 GiB       115.98 GiB     288.5 MiB    0 bytes      294.52 MiB
SWAP     0 bytes        0 bytes        0 bytes        n/a          n/a          n/a
```

To stop a command running in watch mode, press **Ctrl** + **C** to interrupt the process.

### Managing RBFS Services

#### Brick Daemons

RBFS microservices architecture allows daemons to serve various complementary functions and provide services.

For example, the subscriber daemon (`subscriberd`) manages the current subscriber state and is responsible for authentication, authorization, and accounting. The `ribd` daemon is responsible for route selection, next-hop resolution, tunnel selection and recursion.

There are daemons such as `CtrlD` (Controller) and `ApiGwD` (API Gateway) which are part of the RBFS ecosystem. These daemons sit in the middle (on the RtBrick Host) and manage all the communication between the client and backend services running in the container. The API Gateway (`ApiGwD`) daemon provides a single point access to expose services running inside of the RBFS container.

RBFS daemons and other dependencies are packaged as a Debian 12 Bookworm container. The RBFS container is hosted on the RtBrick Host, an open-source operating system, which can be run on white box switches.

RBFS can perform various roles such as Spine, Leaf, and Multiservice Edge which serve different use-cases. The software images of these various roles contain daemons that are required to serve these roles for their different functions. The RBFS Multiservice Edge software image contains all the RBFS daemons packaged in a container, other roles such as Spine and Leaf include only the daemons which are required to carry out their respective functions.

For example, the Spine RBFS image includes (in addition to other daemons) the interior gateway protocol daemons such as `isis.appd`, `isis.iod`, `ospf.appd`, and `ospf.iod` which are not required in the Access Leaf image.

Similarly, the Access Leaf image should include daemons (in addition to other daemons) such as `subscriberd`, `l2tpd`, `pppoed`, and `ipoed` which are not present in the Spine image.

The daemons such as `alertmanager`, `confd`, `etcd`, `fibd`, `hostconfd`, `ifmd` and so on are present in the images of both the Spine and Leaf roles as these daemons are required in both of these roles.

#### Launching Microservices Dynamically

When the RBFS container starts up, it starts only relevant microservices depending on the image role and platform. This is done to minimize unnecessary resource consumption. In RBFS, there are base microservices and on-demand microservices. RBFS containers will have all microservices installed according to the platform and image role, but not all will be enabled on bootup. Only the base microservices will be enabled and started on bootup. On-demand microservices will only be started when their respective configurations are configured and will stop once all dependent configurations are deleted.

For instance, when the user configures BGP with the CLI command `set instance <instance> protocol bgp local-as <as-number>`, the rtbrick-bgp.appd.1 and rtbrick-bgp.iod.1 services will start. And, once the BGP configuration is deleted, "rtbrick-bgp.appd.1" and "rtbrick-bgp.iod.1" will be stopped after 5 minutes (graceful shutdown time).

By default, the following base microservices will be running in the container.

- rtbrick-confd
- rtbrick-etcd
- rtbrick-fibd
- rtbrick-hostconfd
- rtbrick-ifmd
- rtbrick-lldpd
- rtbrick-mribd
- rtbrick-opsd
- rtbrick-poold
- rtbrick-resmond
- rtbrick-resmond-agent
- rtbrick-restconfd
- rtbrick-ribd
- rtbrick-staticd

When you make other RBFS configurations, the required on-demand microservices will be automatically enabled.

#### Listing All Available Services

To list all available RBFS services, enter the following command:

```
sudo systemctl list-unit-files | grep "rtbrick-"
```

#### Starting or Stopping the Services

To start or stop a specific RBFS service, enter the following command:

```
sudo service <service> start|stop
```

#### Restarting a Service

To restart a specific RBFS service, enter the following command:

```
sudo service <service> restart
```

Example:

```
supervisor@onl>multiservice-edge.rtbrick.net:~ $ sudo service rtbrick-ctrld restart
```

- In the unlikely event of a failure in a daemon, the service will restart automatically.

#### Verifying the Status of a Service

To verify the status of a RBFS Service, enter the following command:

```
sudo service <service> status
```

For example, to check the status of the `rtbrick-pppoed.1` service, enter the following command:

```
supervisor@onl>multiservice-edge.rtbrick.net:~ $ sudo service rtbrick-pppoed.1 status
```

#### Removing the Committed/Uncommitted Configuration

To discard the both committed and uncommitted configurations, enter the following command:

```
supervisor@rtbrick>LEAF01: cfg> discard all
```

Exeuting the `discard all` command will remove your committed configurations. Users will receive a warning indicating that it will permanently remove all committed configurations. A confirmation prompt shows the message providing an explicit [yes/no] option for the user to choose before proceeding.

Example:

```
supervisor@rtbrick.net: cfg> discard all
Warning: This action will discard all committed configurations.
Are you sure you want to proceed? [yes/no] [Default: yes]: no
supervisor@rtbrick.net: cfg> commit
```

### Filtering Output using Pipe (|) in Show Commands

RBFS supports output filtering according to the criteria specified by the options using pipe (|) for all show commands.

> Only a single-level pipe with option is supported.

- The pipe (|) command is supported only with show commands.
- The supported two options are: `grep` and `count`.

**Syntax:**

- <show command> | grep <search pattern>
- <show command> | count

Example:

```
supervisor@rtbrick.net: op> show version
UUID        : 4a2b63df-305b-4d15-8729-9cf6c57cb628
Version     : 25.1.0-g6daily.20250123075734+Bdevelopment.Cfbee306f
Role        : multiservice-edge
Platform    : q2a
Format      : lxd
Build date  : 2025-01-23 07:57:34 UTC
supervisor@rtbrick.net: op> show version | grep "Format|Role"
Role        : multiservice-edge
Format      : lxd
supervisor@rtbrick.net: op> show version | count
  <cr>
supervisor@rtbrick.net: op> show version | count
6
```

#### Error Path Information in RESTCONF Configurations

You can see the error path information, which indicates the exact hierarchy of the problematic node within a module in the RESTCONF configurations. It enables troubleshooting easier by providing references to the interface, peer group, instance, or configuration that is causing the problem.

Example:

```
supervisor@testFix>srv02-tst.r1.nbg.rtbrick.net: cfg> show config set time-series
supervisor@testFix>srv02-tst.r1.nbg.rtbrick.net: cfg> set time-series metric test
supervisor@testFix>srv02-tst.r1.nbg.rtbrick.net: cfg> show diff set
+set time-series metric test
supervisor@testFix>srv02-tst.r1.nbg.rtbrick.net: cfg> commit
[lime]Jan 27 10:07:19: Commit failed. Edit and try again or discard changes: application missing-element Mandatory
variable 'table-name' missing for node 'metric' with path: time-series/metric/ <bad-element>table-name</bad-element>
```

#### Supported Platforms

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

## 1.2. BDS Overview

### 1.2.1. BDS Overview

The Brick Data Store (BDS) is a purpose-built, in-memory state database optimized for cloud networking. In RBFS, all system state information is stored as objects in BDS tables. Objects are entries in BDS tables that represent a state.

#### Pub/Sub Model

All Brick Daemons (BD) independently publish and subscribe to tables in a pub/sub model. This model provides resilience and scalability. The figure illustrates the concept:

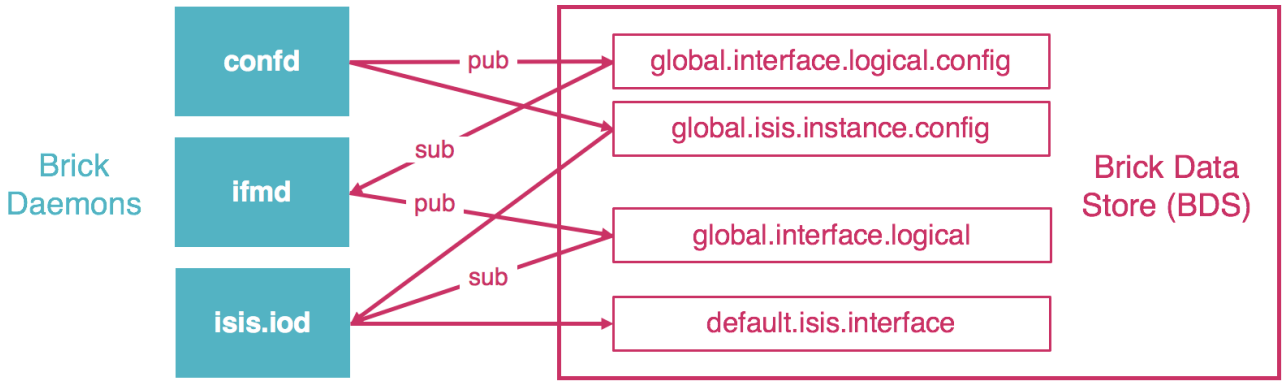



*Figure 1. Pub/Sub Model*

In this example, the configuration daemon (confd) publishes tables that contain configuration data for logical interfaces or IS-IS instances. The interface management daemon (ifmd) is responsible for creating and maintaining interfaces. It therefore subscribes for example to the logical interface configuration table. After processing the data, it creates the logical interfaces and publishes them in the logical interface table. The IS-IS input/output daemon (isis.iod) subscribes to the logical interface table as well as the IS-IS configuration tables. It in turn creates and runs interfaces on which IS-IS protocol packets are exchanged, and publishes them in the IS-IS interface table.

#### BDS User Interface

All BDS tables and objects are fully accessible to the RBS user both via CLI and an API. This provides unprecedented visibility into the system state. This guide covers the BDS CLI. For the BDS API, refer to the BDS API Reference.

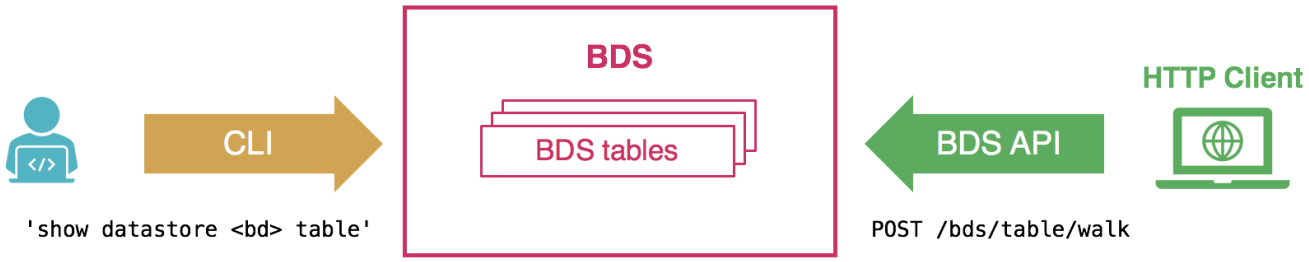



*Figure 2. BDS User Interface*

Please note the RBFS CLI supports show commands to verify the configuration and operation of the system for all features. Therefore you usually do not need to inspect BDS tables directly. For example, for verifying the status of the logical interfaces, you can simply use the 'show interface summary' or the 'show interface logical' commands, instead of displaying the logical interface table. The BDS CLI and API to inspect BDS tables are rather available in addition to advanced analysis or troubleshooting.

#### Supported Platforms

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

### 1.2.2. BDS Operational Commands

This section summarizes some useful BDS CLI commands. It assumes you have some basic knowledge of BDS, and are familiar with the respective tables you are looking for. Describing all tables involved in a particular feature or functionality is out of the scope of this guide.

#### BDS Summary

The BDS summary command provides some metadata of the BDS tables.

**Syntax:**

**show datastore** <bd-name> **summary** <option>

| Option | Description |
|---|---|
| `<bd-name>` | Name of the brick daemon to request this information from. As all BDs independently publish and subscribe to BDS tables, they all hold a different set of tables. As a best practice, select the BD that owns the respective table you are looking for. |
| `table <table-name>` | Display metadata for the given table. |

Example:

```
supervisor@rtbrick>LEAF01: op> show datastore ribd summary
Brick Datastore Summary:
Table Name: local.bds.table.registry.ribd
```

```
  Index                                          Type                Active   Obj
Memory   Index Memory
  sequence-index                                 bds_rtb_bplus          234   24.38
KB     9.59 KB
  gc-index                                       bds_rtb_bplus            0   0
bytes      0 bytes
  table-name-index                               bplus                  234   24.38
KB     9.59 KB
Table Name: local.trim.qrunner.table
  Index                                          Type                Active   Obj
Memory   Index Memory
  sequence-index                                 bds_rtb_bplus            7   840
bytes    728 bytes
  gc-index                                       bds_rtb_bplus            0   0
bytes      0 bytes
  immutable_index                                bplus                    7   840
bytes    728 bytes
  qrunner-index                                  qrunner                  7   840
bytes    728 bytes
Table Name: local.bds.statistics
  Index                                          Type                Active   Obj
Memory   Index Memory
  sequence-index                                 bds_rtb_bplus          319   34.84
KB     14.36 KB
  gc-index                                       bds_rtb_bplus            0   0
bytes      0 bytes
  immutable-index                                bplus                  319   34.84
KB     14.36 KB
Table Name: local.bds.module.registry
  Index                                          Type                Active   Obj
Memory   Index Memory
  sequence-index                                 bds_rtb_bplus           68   3.53
KB      2.79 KB
  gc-index                                       bds_rtb_bplus            0   0
bytes      0 bytes
  module-name-index                              bplus                   68   3.53
KB      2.79 KB
<...>
```

**BDS Tables**

You can use the BDS table commands to display the table objects that contain the actual state information.

**Syntax:**

**show datastore** <bd-name> **table** <option>

| Option | Description |
|---|---|
| `<bd-name>` | Name of the brick daemon to request this information from. As all BDs independently publish and subscribe to BDS tables, they all hold a different set of tables. As a best practice, select the BD that owns the respective table you are looking for. |
| `<table-name>` | Name of the BDS table to display. Without further options, this command displays all objects in a table format. |
| `<table-name> json` | Display the complete table data in JSON format. |
| `<table-name> attribute <attribute-name> <attribute-value> exact` | Filter the table objects based on attribute name and value. You can filter on any attribute, except for attributes of type array. The filter performs a regex match. You can therefore specify the attribute value as a regular expression (regex). You can use the exact match along with the (default) regular expression match. |
| `<table-name> summary` | Display metadata for the given table. |
| `properties` | Display owner, published/subscribed, and locality information for all tables known by the given daemon. |

Example 1: Logical Interface Table

```
supervisor@rtbrick>LEAF01: op> show datastore ifmd table global.interface.logical
Object: 0, Sequence 100125, Last update: Mon Apr 03 13:49:39 GMT +0000 2023
  Attribute                                Type                           Length
Value
  logical_unit_id (1)                      uint16 (3)                          2     0
  ifl_name (2)                             string (9)                         13
ifl-0/1/31/0
  ifp_name (3)                             string (9)                         11
ifp-0/1/31
  instance (5)                             string (9)                          8
default
  mac_address (8)                          macaddr (22)                        6
e8:c5:7a:8f:76:f2
  ipv4_status (10)                         uint8 (2)                           1
up
  ipv6_status (12)                         uint8 (2)                           1
up
  mpls_mtu (13)                            uint16 (3)                          2
1500
  mpls_status (14)                         uint8 (2)                           1
up
```

```
  iso_mtu (15)                            uint16 (3)                          2
1500
  iso_status (16)                         uint8 (2)                           1
down
  admin_status (17)                       uint8 (2)                           1
up
  link_status (18)                        uint8 (2)                           1
up
  ifl_type (19)                           uint8 (2)                           1
Logical Sub interface
  operational_status (24)                 uint8 (2)                           1
up
  ifindex (25)                            uint32 (4)                          4
63745
  instance_id (27)                        uint32 (4)                          4     0
<...>
```

Example 2: Filter IPv6 Route Table by Prefix

```
supervisor@rtbrick>LEAF01: op> show datastore ribd table default.ribd.1.fib-
local.ipv6.unicast
 attribute prefix6 2001:db8::1/128
Object: 0, Sequence 1900002, Last update: Mon Apr 03 13:49:39 GMT +0000 2023
  Attribute                               Type                            Length
Value
  prefix6 (4)                             ipv6prefix (16)                    17
2001:db8::1/128
  source (11)                             uint8 (2)                           1
direct
  sub_src (12)                            uint8 (2)                           1
Host
  nexthop_key (25)                        payload (8)                        24
3c86eaebe6617cf61ed96c819cfd63839bd90cc85a533067
  preference (40)                         uint32 (4)                          4     0
  bcm_status (52)                         uint8 (2)                           1
None
  return_code (53)                        uint32 (4)                          4     0
  vpp_status (54)                         uint8 (2)                           1
None
  route_status (55)                       uint32 (4)                          4
|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-
```

Example 3: Filter IPv6 Route Table with Exact Match

```
supervisor@rtbrick>LEAF01: op> show datastore bgp.appd.1 table vrf1.bgp.rib-
in.ipv4.unicast.198.51.100.30.198.51.100.25 attribute prefix4 198.51.100.11/24 exact
Object: 0, Sequence: 367772, Last update: Wed May 19 08:05:08 GMT +0000 2021
  Attribute                               Type                            Length
Value
```

```
  status (1)                              uint8 (2)                          1
Valid
  recv_path_id (2)                        uint32 (4)                         4     0
  prefix4 (3)                             ipv4prefix (13)                    5
198.51.100.40/24
  rd (5)                                  route-distinguisher (40)           8
198.51.100.100:65001
  source (6)                              uint8 (2)                          1
bgp
  sub_src (7)                             uint8 (2)                          1
Local-Peer
  as_path (9)                             array (7), uint32 (4)             20
[57381, 42708, 1299, 5511, 3215]
  origin (10)                             uint8 (2)                          1
IGP
  peer_type (12)                          uint8 (2)                          1     2
  igp_metric (13)                         uint32 (4)                         4
4294967295
  send_path_id (18)                       uint32 (4)                         4
3238151775
  bgp_nh4 (19)                            ipv4addr (12)                      4
198.51.100.30
  community (24)                          array (7), community (27)          8
['1299:20000', '42708:200']
```

**BDS Schema**

The Brick Data Store is schema-driven. Table and object schema definitions are located in RBFS in /usr/share/rtbrick/libbds/. Instead of inspecting schema files, you can use the BDS schema commands to view the schemata directly in the CLI.

**Syntax:**

**show datastore** <bd-name> **schema** <option>

| Option | Description |
|---|---|
| `<bd-name>` | Name of the brick daemon to request this information from. As all BDs independently publish and subscribe to BDS tables, they all hold a different set of tables. To view a table or object schema, you can select any BDs that know the respective table. |
| `table-name <table-name>` | Display the schema of the given table. |
| `object object-name <object-name>` | Display the schema of the given object. |
| `object table-name <table-name>` | Display the schema of the object for a given table. This option is useful if you do not know the name of the object but the name of the table in which it is used. |

**BDS Statistics Memory**

The BDS statistics memory command provides detailed memory usage information.

**Syntax:**

```
show datastore <bd-name> statistics memory
```

| Option | Description |
| --- | --- |
| `<bd-name>` | Name of the brick daemon of which to display the memory usage information. |

# 2. Routing

The Routing section of the RtBrick FullStack (RBFS) User Guides contains comprehensive configuration and operational commands for essential network layer protocols.

Key Features:

- Details configuration and operations for standard interior gateway protocols, specifically IS-IS and OSPFv2/v3, to manage internal network topology discovery.
- Covers BGP configurations alongside mitigation features like BGP FlowSpec for traffic filtering and BGP RPKI-RTR for route origin validation.
- Explains how to implement the Label Distribution Protocol (LDP) to build and signal MPLS transport tunnels across the network core.
- Details the management of the central Routing Information Base (RIB) and the setup of Static Routing for fixed paths.
- Outlines how to control traffic movement using specific routing Policies and how to share paths between different routing tables via Route Leaking.

The following guides available under the Routing section:

- RIB
- Route Leaking
- BGP
- BGP FlowSpec
- BGP RPKI-RTR Route Validation
- IS-IS
- OSPFv2/v3
- LDP
- Policy
- Static Routing

## 2.1. RIB

### 2.1.1. RIB Overview

A Routing Information Base (RIB) stores and maintains the route information. The RIB table contains information about each reachable network prefix and the next hop information. Each routing protocol inserts its routes into the RIB when it learns a new route. The RBFS route manager (ribd) selects the best routes from the protocol RIBs and places them into the Forwarding Information Base (FIB) table. If a destination becomes unreachable, the route is marked unusable and eventually removed from the RIB

The main differences between RIB and FIB are:

- RIB is about storing and maintaining route information, including attributes for route selection (control plane), etc.
- FIB is used to determine how packets are forwarded (data plane), i.e., only contains a subset of information from RIB. Also, FIB is downloaded to ASIC or VPP.

The image below shows how various routing protocols insert their routes into the RIB.

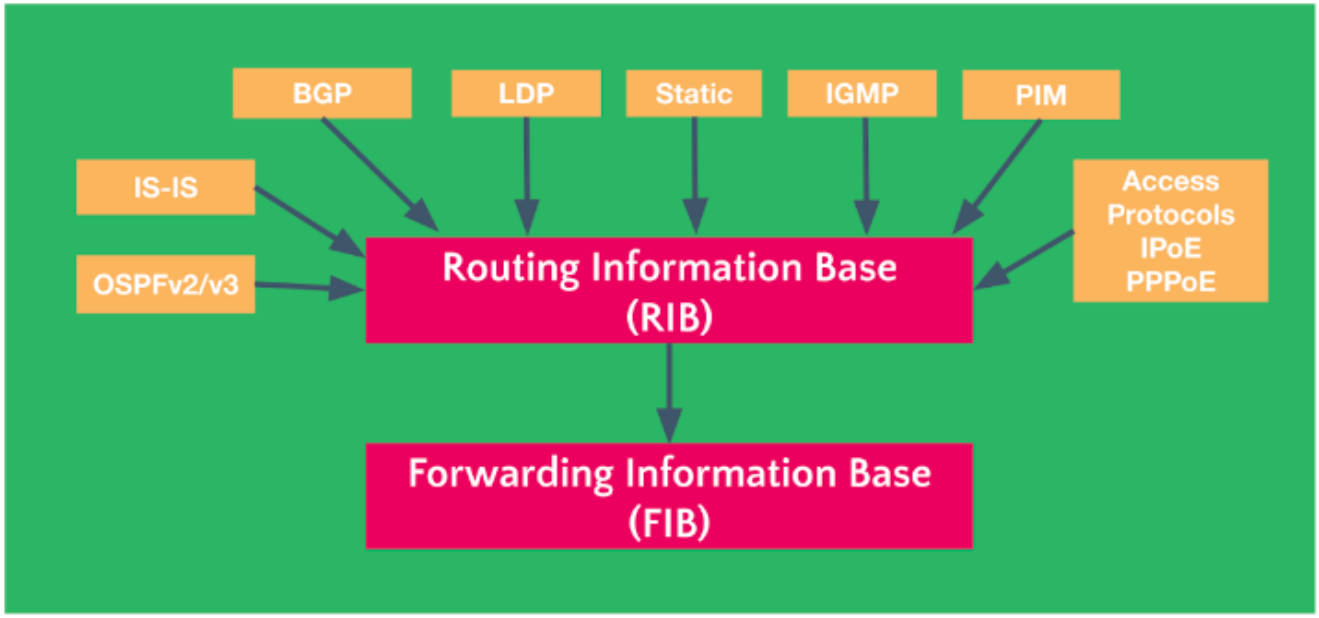



*Figure 3. RIB Overview*

**Routing Instances**

A routing instance represents a set of interfaces, routing protocols, and corresponding routing tables. You can think of a routing instance as a virtual router within RBFS.

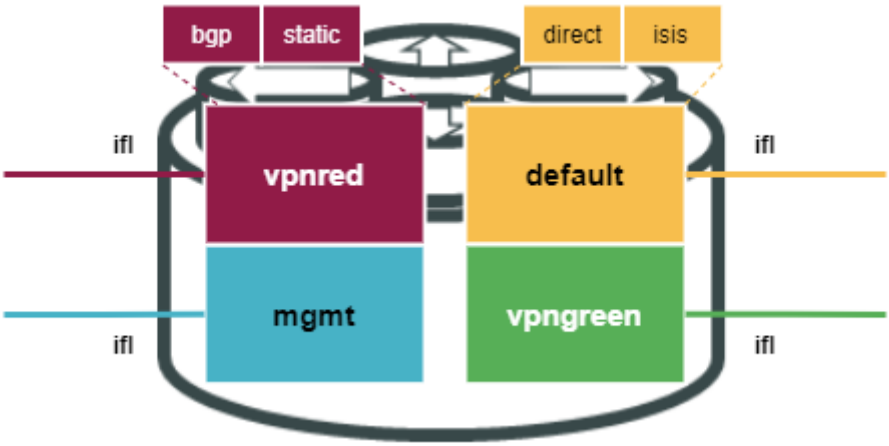



*Figure 4. Routing Instances*

Each logical interface (ifl) is associated uniquely with a routing instance. If no routing instance is explicitly configured, the default routing instance is used. The interface association can be seen with the `show interface address` command.

Routing instances are configured using the set instance <instance_name> syntax. When configuring a routing instance, you must specify which address families (AFI, SAF) the routing instance should support.

### Route Preference

Several routing protocols, including static routes, may provide multiple paths to reach a particular destination. However, not all of these paths are necessarily optimal. At any given time, only one routing protocol can identify the most optimal route to the destination. Each routing protocol, including static routes, is given a preference value, where lower values indicate higher preference. In cases where multiple routing sources are present, the route identified by the routing protocol with the highest preference is selected as the best option and added to the local routing table.

| Routing Protocol | Route Preference |
|---|---|
| Direct | 0 |
| Static | 2 |
| IPoE | 7 |
| PPP | 8 |
| LDP | 9 |
| OSPF | 10 |
| IS-IS LEVEL 1 | 15 |
| IS-IS LEVEL 2 | 18 |
| eBGP | 20 |
| iBGP | 200 |

### Route Prioritization

Routes are categorized into high and low priority for resolution and installation.

- **High Priority Routes include:**
  - Static Routes
  - Direct
  - ARP/ND
  - IGP /32 or /128 Unicast
  - Labeled Unicast /32 or /128

All Other Routes are considered as low priority routes.

High priority routes are processed before all other low priority routes during route resolution and installation.

### Route Selection

RIBD obtains routes and corresponding next-hops from various sources such as static, protocols, and direct connections. Route selection and resolution are also handled by RIBD, which chooses the best route based on the preference assigned to each source. If multiple sources provide the same prefix, RIBD will select the best route based on the source’s preference and install it in FIBD.

### Route Resolution

Route resolution is the process of finding the forwarding next hop from protocol nexthop downloaded into RIBD. For this purpose RIBD performs recursive route lookups till it finds a direct outgoing interface for the route.

For example, BGP installs a route with nexthop set to its peer IP address. However, it doesn't provide the outgoing interface. So, RIBD will check the peer IP address (next-hop IP) in its routing table (most probably downloaded by IGPs) and find the directly connected router's IP address and its outgoing nexthop.

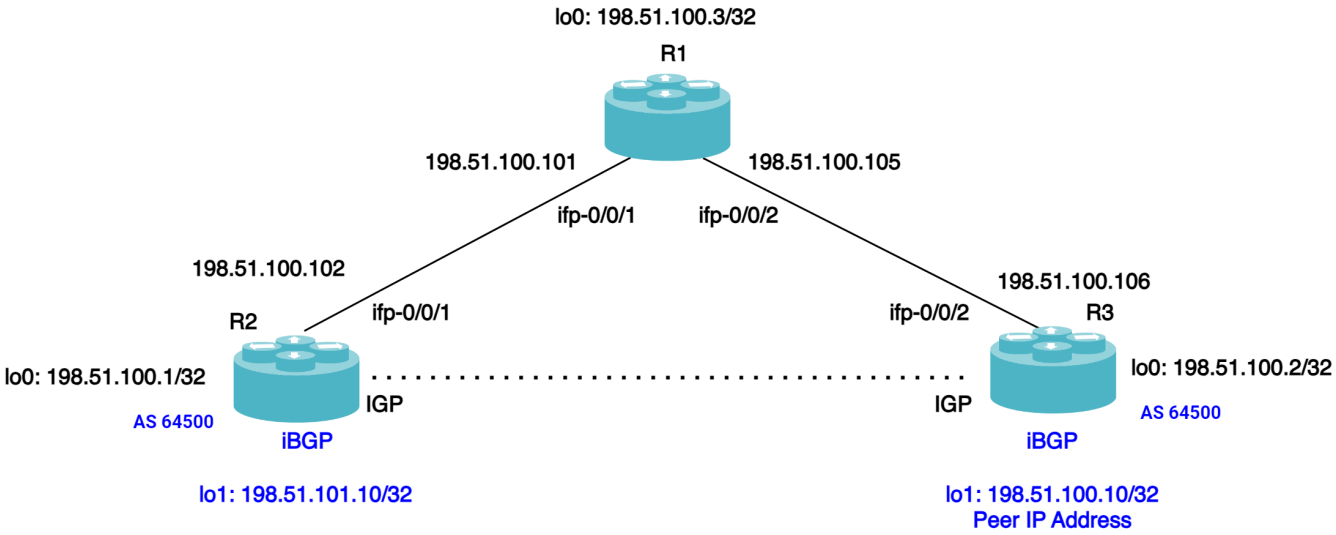



*Figure 5. RIB Route Resolution*

The BGP route for 198.51.100.10/32 will only be added to the routing table of router R2 if the IP address listed as the next-hop attribute is already reachable based on the information stored in the routing table. Additionally, the BGP route that is installed will contain a reference to the next-hop address 198.51.100.2 (refer to the figure above).

The network 198.51.100.10/32 can be accessed through an IP address that is not directly connected. The physical interface is not located where the BGP route is installed, so it is added to the IP routing table without any information about the outgoing interface. To find the BGP next-hop in the routing table, the router must perform a recursive lookup. However, to use a BGP route, the BGP next-hop IP address must be reachable. This is usually done through an IGP that provides reachability information. In this case, the BGP next-hop 198.51.100.2 is found in the routing table of R2, which is known via OSPF. The outgoing interface is ifp-0/0/1.

During the first route lookup, the router checks whether the destination prefix is in the routing table. If it is, then a recursive lookup is performed for the next-hop IP address. Since the next hop address is not a directly connected interface, the router needs to do a recursive lookup to find it.

Below is an example of the BGP route resolution for prefix:198.51.100.10/32:

```
supervisor@rtbrick.net: op> show route-resolution resolved prefix 198.51.100.10
198.51.100.10, Source: bgp
  Destination instance: default, AFI: ipv4, SAFI: unicast
  Lookup      instance: default, AFI: ipv4, SAFI: unicast
```

```
Covering Prefix: 198.51.100.10
  Interface        MAC Address         Nexthop
  ifl-0/0/1/13    e8:c5:7a:8f:56:47    198.51.100.101
```

**Policy Attachments**

Routing Policies are a set of rules that enable you to manage and alter the default behavior of routing protocols, like BGP and IS-IS. Such policies consist of various "terms," which include "match" and "action" sections with control. The traffic that matches the "match" block is handled by the "action" block.

Once policies have been created, they need to be applied to take effect. Attachment points describe the specific applications and processes to which policies can be applied.

For more information, see section “2.2.4. Attaching Policies” of the Policy User Guide.

**Nexthops**

Nexthop helps routers determine the best path for forwarding data packets to their final destination efficiently. The next hop is identified by its IP address, which is stored in the routing table alongside the associated network prefix and other routing information.

**Adjacency**

The FIB learns the routing information from the routing table and tracks the next hop for all routes. The adjacency table maintains Layer 2 information(Nexthop) for the routes listed in the FIB (the resolved routes that can be installed into the hardware).

**FIB**

FIB determines how packets are forwarded (data plane), i.e., it only contains a subset of information from RIB. Also, FIB is downloaded to ASIC or VPP.

The main difference between the RIB and the FIB is that the RIB contains all the routes the router has learned, while the FIB only contains the best paths to each destination network. The RIB is constantly updated by the routing protocols and static configuration, while the FIB is only updated when the router needs to recalculate the best paths. The FIB is also updated more quickly than the RIB since it only needs to store the best paths and not all routes. The FIB maintains next-hop address information based on the information in the IP routing table.

**ECMP Hashing Algorithm**

The ECMP hashing algorithm distributes traffic across multiple paths or links for load balancing. It achieves this by using various packet header fields (like IP addresses, port numbers, or MAC addresses) to generate a hash value, ensuring probability and an equitable distribution of network traffic.

The following table lists the packet header fields that are used by the ECMP hashing algorithm to generate a hash value, which in turn determines how traffic is distributed. A tick (✔) indicates that the field is used by the hashing algorithm. A cross mark (✘) indicates that the field is not used by the

hashing algorithm.

#### IPv4 UDP/TCP

The following table lists the header fields used in the ECMP hashing algorithm for different platforms when handling IPv4 UDP/TCP traffic.

| Header Field | UfiSpace S9600-102XC | UfiSpace S9600-72XC | UfiSpace S9600-32X | UfiSpace S9510-28DC | UfiSpace S9500-22XST | Edgecore AGR420 | Edgecore AGR400 | Edgecore CSR440 |
|---|---|---|---|---|---|---|---|---|
| Source IP Address (SIP) | ✔ | ✔ | ✔ | ✔ | □ | ✔ | ✔ | ✔ |
| Destination IP Address (DIP) | ✔ | ✔ | ✔ | ✔ | □ | ✔ | ✔ | ✔ |
| Protocol | ✔ | ✔ | ✔ | ✔ | □ | ✔ | ✔ | ✔ |
| Source Port | ✔ | ✔ | ✔ | ✔ | □ | ✔ | ✔ | ✔ |
| Destination Port | ✔ | ✔ | ✔ | ✔ | □ | ✔ | ✔ | ✔ |

#### IPv4

The following table lists the header fields used in the ECMP hashing algorithm for different platforms when handling IPv4 traffic.

| Header Field | UfiSpace S9600-102XC | UfiSpace S9600-72XC | UfiSpace S9600-32X | UfiSpace S9510-28DC | UfiSpace S9500-22XST | Edgecore AGR420 | Edgecore AGR400 | Edgecore CSR440 |
|---|---|---|---|---|---|---|---|---|
| Source IP Address (SIP) | ✔ | ✔ | ✔ | ✔ | □ | ✔ | ✔ | ✔ |
| Destination IP Address (DIP) | ✔ | ✔ | ✔ | ✔ | □ | ✔ | ✔ | ✔ |
| Protocol | ✔ | ✔ | ✔ | ✔ | □ | ✔ | ✔ | ✔ |

#### IPv6 UDP/TCP

The following table lists the header fields used in the ECMP hashing algorithm for different platforms when handling IPv6 UDP/TCP traffic.

| Header Field | UfiSpace S9600-102XC | UfiSpace S9600-72XC | UfiSpace S9600-32X | UfiSpace S9510-28DC | UfiSpace S9500-22XST | Edgecore AGR420 | Edgecore AGR400 | Edgecore CSR440 |
|---|---|---|---|---|---|---|---|---|
| Source IP Address (SIP) | ✔ | ✔ | ✔ | ✔ | □ | ✔ | ✔ | ✔ |
| Destination IP Address (DIP) | ✔ | ✔ | ✔ | ✔ | □ | ✔ | ✔ | ✔ |

| Header Field | UfiSpace S9600-102XC | UfiSpace S9600-72XC | UfiSpace S9600-32X | UfiSpace S9510-28DC | UfiSpace S9500-22XST | Edgecore AGR420 | Edgecore AGR400 | Edgecore CSR440 |
|---|---|---|---|---|---|---|---|---|
| Next Header | ✔ | ✔ | ✔ | ✔ | ▯ | ✔ | ✔ | ✔ |
| Source Port | ✔ | ✔ | ✔ | ✔ | ▯ | ✔ | ✔ | ✔ |
| Destination Port | ✔ | ✔ | ✔ | ✔ | ▯ | ✔ | ✔ | ✔ |
| Flow Label (FL) | ✔ | ✔ | ✔ | ✔ | ▯ | ✔ | ✔ | ✔ |

**IPv6**

The following table lists the header fields used in the ECMP hashing algorithm for different platforms when handling IPv6 traffic.

| Header Field | UfiSpace S9600-102XC | UfiSpace S9600-72XC | UfiSpace S9600-32X | UfiSpace S9510-28DC | UfiSpace S9500-22XST | Edgecore AGR420 | Edgecore AGR400 | Edgecore CSR440 |
|---|---|---|---|---|---|---|---|---|
| Source IP Address (SIP) | ✔ | ✔ | ✔ | ✔ | ▯ | ✔ | ✔ | ✔ |
| Destination IP Address (DIP) | ✔ | ✔ | ✔ | ✔ | ▯ | ✔ | ✔ | ✔ |
| Next Header | ✔ | ✔ | ✔ | ✔ | ▯ | ✔ | ✔ | ✔ |
| Flow Label (FL) | ✔ | ✔ | ✔ | ✔ | ▯ | ✔ | ✔ | ✔ |

**Ethernet (L2 Forwarding)**

The following table lists the header fields used in the ECMP hashing algorithm for different platforms when handling Ethernet (L2 Forwarding) traffic.

| Header Field | UfiSpace S9600-102XC | UfiSpace S9600-72XC | UfiSpace S9600-32X | UfiSpace S9510-28DC | UfiSpace S9500-22XST | Edgecore AGR420 | Edgecore AGR400 | Edgecore CSR440 |
|---|---|---|---|---|---|---|---|---|
| Source MAC | ✔ | ✔ | ✔ | ✔ | ▯ | ✔ | ✔ | ✔ |
| Destination MAC | ✔ | ✔ | ✔ | ✔ | ▯ | ✔ | ✔ | ✔ |
| EtherType | ✔ | ✔ | ✔ | ✔ | ▯ | ✔ | ✔ | ✔ |
| VLAN | ✔ | ✔ | ✔ | ✔ | ▯ | ✔ | ✔ | ✔ |

**MPLS**

The following table lists the header fields used in the ECMP hashing algorithm for different platforms when handling MPLS traffic.

| Header Field | UfiSpace S9600-102XC | UfiSpace S9600-72XC | UfiSpace S9600-32X | UfiSpace S9510-28DC | UfiSpace S9500-22XST | Edgecore AGR420 | Edgecore AGR400 | Edgecore CSR440 |
|---|---|---|---|---|---|---|---|---|
| All MPLS Labels | ✔ | ✔ | ✔ | ✔ | □ | ✔ | ✔ | ✔ |

**Field Definitions**

**SIP/DIP (Source/Destination IP Address)**

Identifies the source/destination of the packet, that is, IPv4/IPv6

**Next Header/Protocol**

Identifies the type of payload, such as TCP, UDP etc.

**L4 Ports (Source/Destination Port)**

Used in TCP/UDP headers to identify application endpoints

**FL (Flow Label)**

Used for IPv6 to improve load balancing

**MAC Addresses (Source/Destination MAC)**

Used in Ethernet frames for identifying source and destination devices

**EtherType**

Indicates the protocol encapsulated in the Ethernet frame

**VLAN**

Identifies the VLAN to which the frame belongs

**MPLS Labels**

Used in MPLS networks to identify the destination / Forward Equivalence Class for packet forwarding

### 2.1.2. RIB Operational commands

**Display Routes**

The show route command displays information on routes.

**Syntax:**

```
show route <option>
```

| Option | Description |
|---|---|
| `-` | Without any option, the command shows the information for all routes for all instances. |
| `detail` | Shows detailed route information. |

| Option | Description |
|---|---|
| import | The commands show all the routes imported from another instance. For information about route import command options, see section Show Route Import. |
| incoming interface <name> | Shows the information of the specified interface through which the routes enter to the router. |
| instance <name> | Shows routing table information for a specified instance. |
| hierarchy | Shows routing information in a hierarchical |
| ipv4 | Shows route information for the IPv4 routing table. |
| ipv6 | Shows route information for the IPv6 routing table. |
| l2vpn | Shows route information for the L2VPN routing table. For information about l2vpn command options, see section Show Route L2VPN. |
| mpls | Shows route information for the MPLS routing table. |
| prefix <value> | Shows route information for a specified destination prefix. |
| prefix-length-distribution | Shows the number of routes with the same prefix length for the sources. |
| service | Shows the number of routes with the same service label. |
| source | Shows routes from a specified source. |
| summary | Shows the number of routes selected by RIB for each source. |
| summary platform | Shows BCM status counts for hardware platforms and VPP status counts for virtual platforms. The platform option is available wherever summary option is supported for the show route command. |

Example: The following example shows route information.

```
supervisor@rtbrick: op> show route
Flags: & - Imported
Instance: ce1-vrf, AFI: ipv4, SAFI: unicast
Flags  Prefix/Label                                Source              Pref  Next Hop
Outgoing Interface
       198.51.100.201/32                           direct                 0  198.51.100.201
lo-0/0/3/0
       198.51.100.202/32                           direct                 0  198.51.100.202
lo-0/0/3/1
       198.51.100.221/32                           bgp                   20  198.51.100.21
ifl-0/0/2/0
       198.51.100.222/32                           bgp                   20  198.51.100.21
ifl-0/0/2/0
Instance: ce1-vrf, AFI: ipv6, SAFI: unicast
Flags  Prefix/Label                                Source              Pref  Next Hop
Outgoing Interface
       2001:db8:0:1::201/128                       direct                 0  2001:db8:0:1::201
lo-0/0/3/0
       2001:db8:0:1::202/128                       direct                 0  2001:db8:0:1::202
lo-0/0/3/1
       2001:db8:0:1::221/128                       bgp                   20  198.51.100.21
```

```
ifl-0/0/2/0
       2001:db8:0:1::222/128                      bgp                   20  198.51.100.21
ifl-0/0/2/0
Instance: default, AFI: ipv4, SAFI: unicast
Flags  Prefix/Label                               Source               Pref  Next Hop
Outgoing Interface
       192.0.2.1/32                               direct                  0  192.0.2.1
ifl-0/0/2/0
       192.0.2.0/30                               direct                  0  192.0.2.0
ifl-0/0/2/0
       192.0.2.2/32                               arp-nd                  6  192.0.2.2
ifl-0/0/2/0
       192.0.2.5/32                               direct                  0  192.0.2.5
ifl-0/0/2/100
       192.0.2.4/30                               direct                  0  192.0.2.4
ifl-0/0/2/100
       198.51.100.11/32                           direct                  0  198.51.100.11
lo-0/0/0/0
       198.51.100.12/32                           direct                  0  198.51.100.12
lo-0/0/0/1
       198.51.100.21/32                           bgp                    20  192.0.2.2
ifl-0/0/2/0
       198.51.100.22/32                           bgp                    20  192.0.2.2
ifl-0/0/2/0
Instance: default, AFI: ipv4, SAFI: labeled-unicast
Flags  Prefix/Label                               Source               Pref  Next Hop                      Label
Outgoing Interface
       192.0.2.1/32                               direct                  0  192.0.2.1                     -
ifl-0/0/2/0
       192.0.2.0/30                               direct                  0  192.0.2.0                     -
ifl-0/0/2/0
       192.0.2.2/32                               arp-nd                  6  192.0.2.2                     -
ifl-0/0/2/0
       192.0.2.5/32                               direct                  0  192.0.2.5                     -
ifl-0/0/2/100
       192.0.2.4/30                               direct                  0  192.0.2.4                     -
ifl-0/0/2/100
       198.51.100.11/32                           direct                  0  198.51.100.11                 -
lo-0/0/0/0
       198.51.100.12/32                           direct                  0  198.51.100.12                 -
lo-0/0/0/1
       198.51.100.21/32                           bgp                    20  192.0.2.2                     -
ifl-0/0/2/0
       198.51.100.22/32                           bgp                    20  192.0.2.2                     -
ifl-0/0/2/0
Instance: default, AFI: ipv6, SAFI: unicast
Flags  Prefix/Label                               Source               Pref  Next Hop
Outgoing Interface
       2001:db8:0:1::11/128                       direct                  0  2001:db8:0:1::11
lo-0/0/0/0
       2001:db8:0:1::12/128                       direct                  0  2001:db8:0:1::12
lo-0/0/0/1
       2001:db8:0:1::21/128                       bgp                    20  192.0.2.2
ifl-0/0/2/0
       2001:db8:0:1::22/128                       bgp                    20  192.0.2.2
ifl-0/0/2/0
       2001:db8:1:1::1/128                        direct                  0  2001:db8:1:1::1
ifl-0/0/2/0
       2001:db8:1:1::/64                          direct                  0  2001:db8:1:1::
ifl-0/0/2/0
       2001:db8:2:1::1/128                        direct                  0  2001:db8:2:1::1
ifl-0/0/2/100
       2001:db8:2:1::/64                          direct                  0  2001:db8:2:1::
ifl-0/0/2/100
Instance: default, AFI: ipv6, SAFI: labeled-unicast
Flags  Prefix/Label                               Source               Pref  Next Hop                      Label
```

```
Outgoing Interface
       2001:db8:0:1::11/128                      direct                 0  2001:db8:0:1::11              -
lo-0/0/0/0
       2001:db8:0:1::12/128                      direct                 0  2001:db8:0:1::12              -
lo-0/0/0/1
       2001:db8:0:1::21/128                      bgp                   20  192.0.2.2                     -
ifl-0/0/2/0
       2001:db8:0:1::22/128                      bgp                   20  192.0.2.2                     -
ifl-0/0/2/0
       2001:db8:1:1::1/128                       direct                 0  2001:db8:1:1::1               -
ifl-0/0/2/0
       2001:db8:1:1::/64                         direct                 0  2001:db8:1:1::                -
ifl-0/0/2/0
       2001:db8:2:1::1/128                       direct                 0  2001:db8:2:1::1               -
ifl-0/0/2/100
       2001:db8:2:1::/64                         direct                 0  2001:db8:2:1::                -
ifl-0/0/2/100
Instance: default, AFI: mpls, SAFI: unicast
Flags  Prefix/Label                              Source              Pref  Next Hop
Outgoing Interface
       519936,bos:1                              bgp                  170  -
-
       519937,bos:1                              bgp                  170  -
-
       519938                                    bgp                  170  192.0.2.2
ifl-0/0/2/0
       519939                                    bgp                  170  192.0.2.2
ifl-0/0/2/0
       519940,bos:1                              bgp                  170  198.51.100.21
ifl-0/0/2/0
       519941,bos:1                              bgp                  170  198.51.100.21
ifl-0/0/2/0
```

Example: A route summary is shown in the following example.

```
supervisor@rtbrick: op> show route summary
Instance: ce1-vrf
  Source                 Routes
  bgp                         4
  direct                      4
  Total Routes                8
Instance: default
  Source                 Routes
  arp-nd                      2
  bgp                        14
  direct                     24
  Total Routes               40
```

Example: This example shows routes with IPv4 prefixes of the same length

```
supervisor@rtbrick: op> show route prefix-length-distribution
Instance: ce1-vrf
  Prefix Length      Count
            /32          4
           /128          4
            Sum          8
```

```
Instance: default
  Prefix Length      Count
            /30          4
            /32         14
            /64          4
           /128         12
            Sum         34
```

The following example displays a summary of routes across all routing instances, including route sources and platform-specific route programming status counts for BCM.

```
supervisor@rtbrick.net: cfg> show route summary platform
Instance: default
  Source                                  Routes
  arp-nd                                       8
  bgp                                         32
  direct                                      40
  Total Routes                                80

  BCM Status                               Count
  Download success                            40
  Check cross connect table                    8
Instance: new_instance
  Source                                  Routes
  arp-nd                                       4
  direct                                      20
  Total Routes                                24

  BCM Status                               Count
  Download success                            24
Instance: vrf-blue
  Source                                  Routes
  bgp                                          4
  direct                                       4
  Total Routes                                 8

  BCM Status                               Count
  Download success                             8
Instance: vrf1-evpn-vpws
  Source                                  Routes
  bgp                                          1
  Total Routes                                 1

  BCM Status                               Count
  Download success                             0
  Check cross connect table                    1
Instance: vrf1-vpls-vpws
  Source                                  Routes
  bgp                                          3
  Total Routes                                 3
```

```
  BCM Status                                Count
  Download success                              0
  Check cross connect table                     3
Instance: vrf2-evpn-vpws
  Source                                   Routes
  bgp                                           2
  Total Routes                                  2

  BCM Status                                Count
  Download success                              0
  Check cross connect table                     2
Instance: vrf3-evpn-vpws
  Source                                   Routes
  bgp                                           2
  Total Routes                                  2

  BCM Status                                Count
  Download success                              0
  Check cross connect table                     2
```

The following example displays route source statistics and platform-specific route programming status counts for the specified routing instance.

```
supervisor@rtbrick.net: cfg> show route instance default summary platform
Instance: default
  Source                                   Routes
  arp-nd                                        8
  bgp                                          32
  direct                                       40
  Total Routes                                 80

  BCM Status                                Count
  Download success                             40
  Check cross connect table                     8
```

The following example displays a summary of imported routes along with their platform-specific route programming status counts for BCM.

```
supervisor@rtbrick.net: cfg> show route import summary platform
Instance: new_instance
  Source                                   Routes
  arp-nd                                        4
  direct                                       20
  Total Routes                                 24
  BCM Status                                Count
  Download success                             24
```

#### Show Route Hierarchy

The `show route hierarchy` command provides the complete information of a prefix at a given afi, safi, instance from its source, RIBD, and FIBD.

**Syntax:**

**show route hierarchy** <option>

| Option | Description |
|---|---|
| `instance <instance_name>` | Name fof the instance. |
| `<afi>` | To show only the routes for a specific network protocol, such as IPv4 or IPv6. |
| `<safi>` | To view specific types of routes, such as IPv4 unicast or IPv6 multicast,received from neighbors. |
| `<afi> <safi> detail` | Shows information in detail for a specified afi or safi. |
| `<afi> <safi> instance <instance-name>` | Filter route entries for specified instance for a specifc afi and safi. |
| `label` | Filters route entries by a specific MPLS label. |
| `prefix` | Filters route entries by a specified IP prefix. |

Example:

```
supervisor@rtbrick: op> show route hierarchy ipv4 unicast instance default prefix 198.51.100.21/32
bgp:
  prefix: 198.51.100.21/32
  preference: 20
    nexthop set hash: 2e55b7ee0226a90d80b6162cb3fc99b80820d75920227ca2
      nexthop hash: 943dee060e61ee157d00bb93d5405802c3ee1bcf701d4d07
        nexthop4: 192.0.2.2
ribd:
  prefix: 198.51.100.21/32
  preference: 20
  source: bgp
    adjacency set hash: 2e55b7ee0226a90d80b6162cb3fc99b80820d75920227ca2
      nexthop hash: 943dee060e61ee157d00bb93d5405802c3ee1bcf701d4d07
        nexthop4: 192.0.2.2, exit interface: ifl-0/0/2/0
        mac address: 98:19:2c:99:47:2d
fibd:
  prefix: 198.51.100.21/32
  preference:  20
  source: bgp
  route status: |VPP Route ADD|-|-|-|-|-|-|-|-|-|-|-|-|BCM Route ADD|-|-|-|-|-|-|-|BCM Route Adj Fec Update|-
  bcm status: Download success
  vpp status: Add Processed
    adjacency set hash: 2e55b7ee0226a90d80b6162cb3fc99b80820d75920227ca2
      nexthop hash: 943dee060e61ee157d00bb93d5405802c3ee1bcf701d4d07
        nexthop4: 192.0.2.2, exit interface: ifl-0/0/2/0
        mac address: 98:19:2c:99:47:2d
```

#### Show Route Import

**Syntax:**

**show route import** <option>

| Option | Description |
|---|---|
| `import` | The command shows all the routes imported from another instance. |
| `detail` | Shows detailed information for imported routes. |
| `instance` | Shows routing table information for a specific instance for imported routes. |
| `ipv4` | Shows route information for the IPv4 routing table for imported routes. |
| `ipv6` | Shows route information for the IPv4 routing table for imported routes. |
| `source` | Shows source of the routing information for the specified source. |
| `summary` | Shows the number of imported routes selected by RIB for each source. |

#### Show Route L2VPN

**Syntax:**

**show route l2vpn** <option>

| Option | Description |
|---|---|
| `l2vpn` | Without any option, the commands display all the L2VPN routes. |
| `detail` | Shows detailed information for l2vpn routes. |
| `evpn-vpws` | Shows routing table information for the EVPN-VPWS address family. |
| `instance` | Shows L2VPN routing table information for a specific instance |
| `label <label>` | Shows L2VPN routing table information for destination label. |
| `prefix` | Shows L2VPN routing table information for the destination prefix. |
| `source` | Shows source information for the L2VPN routes. |
| `summary` | Shows the number of L2VPN routes selected by RIBD for each source. |
| `vpls-vpws` | Shows the L2VPN routes from VPLS-VPWS address specified. |

### Show Route Resolution

The `show route-resolution` command displays the routes that were requested to be resolved for their nexthops. Otherwise, it shows the route is unresolved.

**Syntax:**

**show route-resolution** <options>

| Option | Description |
|---|---|
| - | Without any option, the command shows the information for all requests and response tables side by side. |
| destination-instance | Shows the information for all requests and responses for a destination instance. |
| look-up instance | Shows lookup instance routes. |
| prefix | Shows routes for prefix 4 or prefix 6. |
| resolved | Shows resolved routes. |
| source | Shows source of requested source. |
| unresolved | Shows unresolved routes. |

Example 1: Below is an example of the route resolution requested by the protocol BGP.

```
supervisor@rtbrick: op> show route-resolution source bgp
Prefix: 192.0.2.2, Source: bgp
  Destination instance: default, AFI: ipv4, SAFI: unicast
  Lookup instance: default, AFI: ipv4, SAFI: unicast
  Covering Prefix: 192.0.2.2/32
  Interface        MAC Address         Nexthop
  ifl-0/0/2/0      98:19:2c:99:47:2d   192.0.2.2
  ifl-0/0/2/0      98:19:2c:99:47:2d   192.0.2.2
Prefix: 192.0.2.2, Source: bgp
  Destination instance: default, AFI: ipv4, SAFI: labeled-unicast
  Lookup instance: default, AFI: ipv4, SAFI: labeled-unicast
  Covering Prefix: 192.0.2.2/32
  Interface        MAC Address         Nexthop
  ifl-0/0/2/0      98:19:2c:99:47:2d   192.0.2.2
  ifl-0/0/2/0      98:19:2c:99:47:2d   192.0.2.2
Prefix: 192.0.2.2, Source: bgp
  Destination instance: default, AFI: ipv6, SAFI: unicast
  Lookup instance: default, AFI: ipv4, SAFI: unicast
  Covering Prefix: 192.0.2.2/32
  Interface        MAC Address         Nexthop
  ifl-0/0/2/0      98:19:2c:99:47:2d   192.0.2.2
  ifl-0/0/2/0      98:19:2c:99:47:2d   192.0.2.2
Prefix: 192.0.2.2, Source: bgp
  Destination instance: default, AFI: ipv6, SAFI: labeled-unicast
```

```
  Lookup instance: default, AFI: ipv4, SAFI: labeled-unicast
  Covering Prefix: 192.0.2.2/32
  Interface        MAC Address         Nexthop
  ifl-0/0/2/0      98:19:2c:99:47:2d   192.0.2.2
  ifl-0/0/2/0      98:19:2c:99:47:2d   192.0.2.2
Prefix: 198.51.100.21, Source: bgp
  Destination instance: default, AFI: ipv4, SAFI: vpn-unicast
  Lookup instance: default, AFI: ipv4, SAFI: labeled-unicast
  Covering Prefix: 198.51.100.21/32
  Interface        MAC Address         Nexthop
  ifl-0/0/2/0      98:19:2c:99:47:2d   192.0.2.2
  ifl-0/0/2/0      98:19:2c:99:47:2d   192.0.2.2
Prefix: 198.51.100.21, Source: bgp
  Destination instance: default, AFI: ipv6, SAFI: vpn-unicast
  Lookup instance: default, AFI: ipv4, SAFI: labeled-unicast
  Covering Prefix: 198.51.100.21/32
  Interface        MAC Address         Nexthop
  ifl-0/0/2/0      98:19:2c:99:47:2d   192.0.2.2
  ifl-0/0/2/0      98:19:2c:99:47:2d   192.0.2.2
```

Example 2: Below is an example of the unresolved route resolution.

```
supervisor@rtbrick>ufi07.q2c.u21.r4.nbg.rtbrick.net: cfg> show route-resolution unresolved
192.168.16.128, Source: radius
  Lookup      instance: inband_mgmt, AFI: ipv4, SAFI: unicast
  Covering Prefix: None, 7 resolution attempts
198.18.73.251, Source: pim
  Lookup      instance: ip2, AFI: ipv4, SAFI: unicast
  Covering Prefix: None, 7 resolution attempts
```

#### Show Route rib-in

The `show route rib-in` command shows all advertised routes' information received and stored in the inbound RIB (Routing Information Base). These commands allow you to view routes based on specific criteria such as instance, source, address family (AFI), subsequent address family identifier (SAFI) and so on.

**Syntax:**

**show route rib-in** <options>

| Option | Description |
|---|---|
| `-` | Without any option, the command shows the information for all routes in the routing information base. |
| `detail` | Provides a detailed view of routes matching the specified prefix in the inbound RIB. |

| Option | Description |
|---|---|
| `<afi>` | To show only the routes for a specific network protocol, such as IPv4 or IPv6. |
| `<safi>` | To view specific types of routes, such as IPv4 unicast or IPv6 multicast, received from neighbors. |
| `instance <instance>` | Shows the routes in the inbound RIB of a specific routing instance. |
| `source <source>` | Shows the routes in the inbound RIB from a specific neighbor. |
| `summary` | Provides a summary view of all routes in the inbound RIB. |

**Show Route Prefix**

The `show route prefix` command shows all advertised routes' information received and stored in the inbound RIB (Routing Information Base).

**Syntax:**

**show route prefix** <options>

| Option | Description |
|---|---|
| `-` | Without any option, the command shows the information for all routes in the routing information base. |
| `prefix <value>` | Specify the prefix. This is to see if a specific route with a particular prefix has been received from a neighbor. |
| `rib-in` | Shows all routes currently stored in the routing information base without any filtering. |
| `detail` | Provides a detailed view of routes matching the specified prefix in the inbound RIB. |

# 2.2. Route Leaking

## 2.2.1. Route Leaking Overview

Route leaking involves exporting a route from one routing instance to another routing instance.

As shown in the figure below, RBFS allows importing routes from the RIB of one instance to another instance. Additionally, the routes can be selectively imported based on the associated policy.

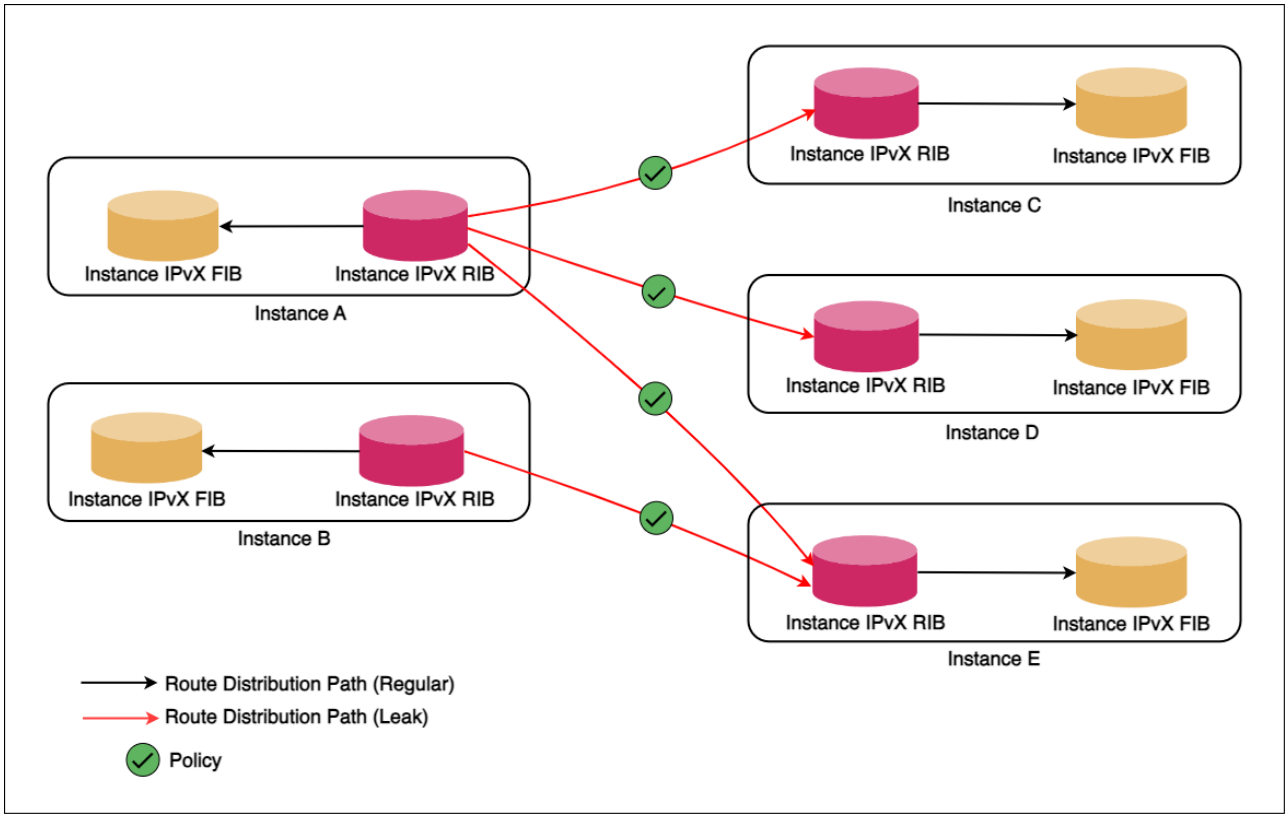



*Figure 6. Route Leaking Across Instances*

### Guidelines and Limitations

- Currently, the route leaking “import-rib” feature is limited to importing only IPv4 and IPv6 unicast routes.

### Supported Platforms

The Route Leaking feature is supported by the Multiservice Edge images for the following Q2A and Q2C hardware platforms.

**Q2A:**

- Edgecore CSR440 (AS7535-28XB)
- UfiSpace S9510-28DC

**Q2C:**

- Edgecore AGR420 (AS7946-74XKSB)
- UfiSpace S9600-72XC
- UfiSpace S9600-102XC

#### Route Leaking Policy Attributes and Match Types

The following table provides all the supported route leaking policy attributes and match types.

| Attribute Type | Match Types Supported |
|---|---|
| ipv4-prefix | regex<br>exact<br>longer<br>or-longer<br>prefix-length-exact<br>prefix-length-greater<br>prefix-length-greater-or-exact |
| ipv6-prefix | regex<br>exact<br>longer<br>or-longer<br>prefix-length-exact<br>prefix-length-greater<br>prefix-length-greater-or-exact |
| community | regex<br>exact<br>exists |
| extended-community | regex<br>exact<br>exists |
| large-community | regex<br>exact<br>exists |
| source | regex<br>exact |

### 2.2.2. Route Leaking Configuration

#### Configuring Route Leaking from one instance to another

To configure route leaking from one instance to another, configure the "import-rib" for the routing instance (AFI/SAFI).

**Syntax:**

**set instance** <destination-instance> **address-family** <afi> <safi> **import-rib** <source-instance>

| Attribute | Value |
|---|---|
| `<destination-instance>` | Name of the routing instance to which routes will be imported to. |
| `<afi>` | Address family identifier (AFI). Supported values: ipv4, ipv6. |
| `<safi>` | Subsequent address family identifier (SAFI). Supported values: unicast. |

| Attribute | Value |
| --- | --- |
| `<source-instance>` | Specifies the routing instance from which the routes will be imported. |

In the following example, IPv4 and IPv6 unicast routes will be copied from the “vrf-blue” to “vrf-red” instance RIB.

```
set instance vrf-red address-family ipv4 unicast import-rib vrf-blue
set instance vrf-red address-family ipv6 unicast import-rib vrf-blue
```

The following is an example of JSON output showing the instance configuration.

```
supervisor@rtbrick>LEAF01: cfg> show config instance vrf-red
{
  "rtbrick-config:instance": [
    {
      "name": "vrf-red",
      "address-family": [
        {
          "afi": "ipv4",
          "safi": "unicast",
          "import-rib": [
            {
              "instance": "vrf-blue"
            }
          ]
        },
        {
          "afi": "ipv6",
          "safi": "unicast",
          "import-rib": [
            {
              "instance": "vrf-blue"
            }
          ]
        }
      ]
    }
  ]
}
```

**Configuring Route Leaking with Policy**

You can attach a policy to import selective routes as per the required criteria.

**Syntax:**

**set instance** <destination-instance> **address-family** <afi> <safi> **import-rib** <source-

```
instance> policy <policy-name>
```

| Attribute | Value |
|---|---|
| `<destination-instance>` | Name of the routing instance to which routes will be imported to. |
| `<afi>` | Address family identifier (AFI). Supported values: ipv4, ipv6. |
| `<safi>` | Subsequent address family identifier (SAFI). Supported values: unicast. |
| `<source-instance>` | Name of the routing instance to which routes will be imported. |
| `<policy-name>` | Name of the routing instance to which routes will be imported based on the specified policy. |

In the following example, the policy `match_ipv4_prefix` is attached to the 'vrf-blue' instance.

```
set instance vrf-red address-family ipv4 unicast import-rib vrf-blue policy
match_ipv4_prefix
```

Example: The following is an example of JSON output of the above configuration.

```
{
  "rtbrick-config:instance": [
    {
      "name": "vrf-red",
      "address-family": [
        {
          "afi": "ipv4",
          "safi": "unicast",
          "import-rib": [
            {
              "instance": "vrf-blue",
              "policy": "match_ipv4_prefix"
            }
          ]
        }
      ]
    }
  ]
}
```

### 2.2.3. Route Leaking Operational Commands

**show route import**

The `show route import` command displays all the imported routes.

**Syntax:**

**show route import** <option>

| Attribute | Value |
|---|---|
| `detail` | Displays detailed route information of the imported routes. |
| `instance <instance-name>` | Displays route information for the imported routes for the specified instance. |
| `instance <instance-name> detail` | Displays detailed route information for the imported routes for the specified instance. |
| `instances <instance-name> prefix <prefix>` | Displays route information for the imported routes for the specified instance and prefix. |
| `instance <instance-name> prefix <prefix> detail` | Displays detailed route information for the imported routes for the specified instance and prefix. |
| `instance <instance-name> summary` | Displays summarized route information for the imported routes for the specified instance. |
| `ipv4` | Displays route information for the IPv4 address family. |
| `ipv6` | Displays route information for the IPv6 address family. |
| `source` | Displays route information for the specified route source. |
| `summary` | Display summarized route information for the imported routes. |

Example 1: Source instance (`vrf-blue`) RIB

```
supervisor@rtbrick: op> show route ipv4 unicast source bgp instance ce1-vrf
Flags: & - Imported
Instance: ce1-vrf, AFI: ipv4, SAFI: unicast
Flags  Prefix/Label                                Source               Pref  Next Hop
Outgoing Interface
       198.51.100.221/32                           bgp                    20  198.51.100.21
ifl-0/0/2/0
       198.51.100.222/32                           bgp                    20  198.51.100.21
ifl-0/0/2/0
```

Example 2: Destination instance RIB

```
supervisor@rtbrick: op> show route import ipv4 unicast source bgp instance ce3-vrf
Flags: & - Imported
Instance: ce3-vrf, AFI: ipv4, SAFI: unicast
Flags  Prefix/Label                                Source               Pref  Next Hop
Outgoing Interface
&      198.51.100.221/32                           bgp                    20  198.51.100.21
ifl-0/0/2/0
&      198.51.100.222/32                           bgp                    20  198.51.100.21
ifl-0/0/2/0
```

Example 3: Detailed view of the destination instance (`vrf-red`) RIB

```
supervisor@rtbrick: op> show route import ipv4 unicast source bgp instance ce3-vrf detail
Instance: ce3-vrf, AFI: ipv4, SAFI: unicast
  Prefix: 198.51.100.221/32
    Source: bgp, Preference: 20
  Import type: inter-instance, Import instance: ce1-vrf
      Next Hop: 198.51.100.21
        Covering prefix: 198.51.100.21/32
        Next Hop type: mpls ingress, Next Hop action: mpls label push
        Resolved in: default-ipv4-labeled-unicast(Primary)
        Outgoing Interface: ifl-0/0/2/0, NextHop MAC: 98:19:2c:99:47:2d
        MPLS-Label: 519936,bos:1
  Prefix: 198.51.100.222/32
    Source: bgp, Preference: 20
  Import type: inter-instance, Import instance: ce1-vrf
      Next Hop: 198.51.100.21
        Covering prefix: 198.51.100.21/32
        Next Hop type: mpls ingress, Next Hop action: mpls label push
        Resolved in: default-ipv4-labeled-unicast(Primary)
        Outgoing Interface: ifl-0/0/2/0, NextHop MAC: 98:19:2c:99:47:2d
        MPLS-Label: 519936,bos:1
```

Example 4: Destination instance (`vrf-red`) RIB after applying the policy (`match_ipv4_prefix`)

```
supervisor@rtbrick: op> show route import ipv4 unicast source bgp instance ce3-vrf
Flags: & - Imported
Instance: ce3-vrf, AFI: ipv4, SAFI: unicast
Flags  Prefix/Label                              Source              Pref  Next Hop
Outgoing Interface
&      198.51.100.221/32                         bgp                   20  198.51.100.21
ifl-0/0/2/0
&      198.51.100.222/32                         bgp                   20  198.51.100.21
ifl-0/0/2/0
```

In the above example, the below policy is used to import the selective route.

```
set policy statement match_ipv4_prefix
set policy statement match_ipv4_prefix ordinal 1
set policy statement match_ipv4_prefix ordinal 1 match
set policy statement match_ipv4_prefix ordinal 1 match rule 1
set policy statement match_ipv4_prefix ordinal 1 match rule 1 type ipv4-prefix
set policy statement match_ipv4_prefix ordinal 1 match rule 1 value-type discrete
set policy statement match_ipv4_prefix ordinal 1 match rule 1 match-type exact
set policy statement match_ipv4_prefix ordinal 1 match rule 1 value 192.168.1.20/32
set policy statement match_ipv4_prefix ordinal 1 action
set policy statement match_ipv4_prefix ordinal 1 action rule 1
set policy statement match_ipv4_prefix ordinal 1 action rule 1 operation return-permit
```

The following is an example of JSON output showing the above policy configuration.

```
{
  "rtbrick-config:statement": [
    {
      "name": "match_ipv4_prefix",
      "ordinal": [
        {
          "ordinal": 1,
          "match": {
            "rule": [
              {
                "rule": 1,
                "type": "ipv4-prefix",
                "value-type": "discrete",
                "match-type": "exact",
                "value": [
                  "192.168.1.20/32"
                  ]
              }
            ]
          },
          "action": {
            "rule": [
              {
                "rule": 1,
                "operation": "return-permit"
              }
            ]
          }
        }
      ]
    }
  ]
}
```

# 2.3. BGP

## 2.3.1. BGP Overview

BGP is a standard exterior gateway protocol (EGP) supported by RtBrick. BGP is considered a “Path Vector” routing protocol and maintains a separate routing table based on the shortest Autonomous System (AS) path and various other route attributes.

### Supported BGP Standards

| RFC Number | Description |
|---|---|
| RFC 2545 | Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing |
| RFC 2918 | Route Refresh Capability for BGP-4 |

| RFC Number | Description |
|---|---|
| RFC 4271 | A Border Gateway Protocol 4 (BGP-4) |
| RFC 4364 | BGP/MPLS IP Virtual Private Networks (VPNs) |
| RFC 4456 | BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP) |
| RFC 4486 | Subcodes for BGP Cease Notification Message |
| RFC 4760 | Multiprotocol Extensions for BGP-4 |
| RFC 5492 | Capabilities Advertisement with BGP-4 |
| RFC 6793 | BGP Support for Four-Octet Autonomous System (AS) Number Space |
| RFC 6608 | Subcodes for BGP Finite State Machine Error |
| RFC 6774 | Distribution of Diverse BGP Paths [Partial Support] |

RFC and draft compliance are partial except as specified.

**Supported Platforms**

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

**Supported BGP Features**

The RBFS supports the following BGP functions:

- Basic BGP Protocol
- Multiprotocol extension for BGP
- Multipath for iBGP and eBGP
- Four-byte AS numbers
- Nexthop Self or next-hop unchanged
- Fast external-failover
- Route reflection
- MD5 Authentication
- Generalized TTL Security Mechanism (GTSM)
- Route Refresh
- Advanced route refresh
- Route redistribution
- Multihop EBGP
- Route selection flexibility (always compare MED, ignore AS Path, and so on)
- Add path
- Hostname/Domain name

- Dynamic peers
- Community, Extended Community, and Large Community support
- 6PE Support
- Curated Upstream Path Selection

The statements and commands required to configure and verify the functioning of BGP features are described in this guide.

#### MD5 Authentication

BGP supports the authentication mechanism using the Message Digest 5 (MD5) algorithm. When authentication is enabled, any Transmission Control Protocol (TCP) segment belonging to BGP exchanged between the peers is verified and accepted only if authentication is successful. For authentication to be successful, both peers must be configured with the same password. If authentication fails, the BGP neighbor relationship is not established.

#### GTSM

The Generalized TTL Security Mechanism (GTSM) protects a BGP session by evaluating the Time to Live (TTL) value contained in the IP header of incoming BGP packets.

TTL security is enabled using the ttl-security command on a single-hop BGP session. When TTL security is enabled on a BGP session the IP TTL values in packets is set to 255. To enable TTL security on a multihop BGP session, enable ttl-security and configure ttl-limit to match the expected TTL value.

For details about configuring GTSM, see the Peer Group Configuration section.

#### IPv6 Provider Edge (6PE)

The Provider Edge (6PE) solution enables IPv6 communication over the MPLS IPv4 core network. IPv6 reachability information is associated with a label and transferred through MP-BGP(AFI: 2 SAFI:4). IPv4 mapped IPv6 address is used to encode the next-hop information. The edge nodes in the MPLS IPv4 core have to support both IPv4 and IPv6. The IPv6 Labeled Unicast routes received from the 6PE peer is considered as IPv6 unicast routes and installed in IPv6 Unicast FIB. The received Label is attached to the IPv6 data traffic at the Ingress node and tunneled through an MPLS tunnel(SR) to the egress node, the label identifies the IPv6 traffic, and the egress node would POP the label and forward the ipv6 traffic towards the destination.

#### Policies

#### The Role of a Routing Policy

Routing Policies are the rules that allow you to control and modify the default behavior of the routing protocols such as BGP and IS-IS. To use routing policies, you configure policies and then apply policies to peer groups or instances.

#### Attachment Points

Policies are useful when they are applied to routes, for which they need to be made known to

routing protocols. In BGP, for example, there are several situations where policies can be used, the most common of these is defining import and export policy. The policy attachment point is the point in which an association is formed between a specific protocol entity, in this case, a BGP neighbor, and a specific named policy.

RtBrick supports attaching a BGP routing policy at two levels:

- Peer group address-family level
- Instance address-family level

In each case, you can apply the policy as an import or export policy and filter. As expected, import filters determine which routing updates are accepted and export filters determine which routes are advertised to other peers.

#### Policy Processing

An import policy, when applied to an address family at the peer group level, examines all *incoming* routes from all BGP peers in the peer group, but only for that address family.

An export policy, when applied to an address family at the peer group level, examines all outgoing routes to all BGP peers in the peer group, but only for that address family.

At the instance level, routing policies that are applied to an address family can work as import or export policies, but for the instances as a whole.

An import policy, when applied to an address family at the instance level, examines all incoming routes before accepting the information only from global or default tables to other instances or VRF tables.

An export policy, when applied to an address family at the instance level, examines all outgoing routes before sending the information from the VRF to global, and then to the VPN table (default).

### BGP Communities

BGP communities are a way to tag routes with additional information that can be used for routing decisions and routing policies. A BGP community is a collection of destinations that share a common property. It allows you to apply actions to that group of destinations as a whole, rather than to each member individually. BGP communities are used to provide an additional capability to tag routes and thereby, modify BGP routing policies.

Communities can influence routing decisions such as accept, reject, preference, and redistribution. RBFS allows various operations such as appending, modifying, or deleting communities as the route travels from one router to another. A BGP community is an optional attribute that can be tagged with a route. The AS path attribute contains community information, which is used to identify community members. It allows for route manipulation without requiring multiple route filters.

RBFS supports three types of BGP communities for implementing complex routing policies and effectively controlling traffic across diverse and large-scale networks.

#### Standard Communities

These are used to control routing policies and are represented by a 32-bit number. They include well-known Communities such as no-export, no-advertise, and so on.

#### Extended Communities

Extended Communities provide functionalities beyond what standard communities offer. They are represented by 64-bit number and are displayed as two 32-bit numbers. They include additional fields such as type and sub-type, providing more granularity and the ability to carry additional information. Types: These include route targets, route origin, and other specialized community types that extend.

#### Large Communities

Represented as a 96-bit number, large communities consist of three 32-bit numbers separated by colons (for example, 64512:100:200). They provide space for tagging routes with extensive information and are easier to use in global policies. Large communities are used in large-scale deployments due to the larger addressing space.

### Support for Allow As In

The Allow AS In feature enables BGP routers to receive and process routes even if the routes carry the router's autonomous system number in the 'AS-Path' attribute. Usually, a BGP router discards routes if the 'AS-Path' attribute contains its autonomous system number to prevent loops. However, with the "Allow AS In" feature, such routes can be accepted and routed to the destination without being rejected. You can specify the allow-as-in option value in the peer configuration. For example, if you specify the value '5', the router accepts BGP routes that contain its own AS number (in the AS path) up to 5 times.

### BGP Best Path Selection Algorithm

BGP routers typically receive multiple paths to the same destination. A BGP router forms a neighbor relationship by connecting to its neighbors and exchanging the routes, once the connection is established. The BGP route selection algorithm decides which is the best path to install in the IP routing table and to use for traffic forwarding.

#### BGP Best Path Selection Algorithm

The algorithm eliminates all routes whose next hop is not reachable. Circular route resolution is considered for route resolution.

The algorithm for determining all the routes that have the same route prefix is as follows:

1. The first route selection is performed based on the lowest route source. Route from the local route source is always preferred over the received route. For example, when there is the same prefix route that is redistributed and received from a neighbor, the local (redistributed route) is always preferred. The locally learned route is preferred over the locally crossed or remote crossed route (in the case of VPN, a route might be learned locally in the VRF. The same prefix might be received from the remote as VPNv4. After importing into the VRF routing table, a locally learned route is preferred over the remote local crossed route).

2. Prefer the path with the highest local preference if the route source is the same. If a path does not have a local preference attribute (for example, it is received from an eBGP peer), then it is considered to have the local preference assigned in the given BGP instance. The `show bgp summary` command shows the local preference assigned in the system. This can be changed using the `set local-preference` value.
3. Prefer the route with the shortest AS path, if no route originated. If there is no `AS_PATH` attribute, then it is assumed to be of length 0. A single `AS_SET` is considered to be a length of `1`.
4. Prefer the path with the lowest origin type, if the AS path length is the same as all the paths. The available three values include `IGP`, `EGP` and `Incomplete`. The lowest value is `IGP` and the highest value is `Incomplete`.
5. Prefer the path with the lowest Multi Exit Discriminator (MED), if the original codes are the same. (By default, MED values are only compared when routes are learned from the same AS. Routes without MED values are treated as if they have a MED value of `0`, which is the lowest and, therefore, always the most preferred value.)
6. Prefer external BGP learned routes over internal BGP routes at this point after comparing the route type (internal BGP and external BGP).
7. Prefer the path whose next hop is resolved through the IGP route with the lowest metric.
8. Prefer the length with a shorter CLUSTER length path. If the CLUSTER attribute is not present, the length is assumed to be `0`.
9. Prefer the path from the peer with the lowest router ID. For any path with an originator ID attribute, substitute the originator ID for the router ID during router ID comparison.
10. Prefer the lowest peer IP address as the tie-breaker, if the router-id is the same for both sessions. This is for BGP to make route selections in case of multiple peerings are used between the same routers.
11. If add path is enabled, then the same peer might advertise multiple paths for the same prefix. The path with a lower send path ID is preferred.

The BGP best path selection algorithm also provides a mechanism to discard paths that are not considered candidates for the best path. The following paths are discarded:

- The paths for which next-hops are not resolved.
- The paths originated from an eBGP neighbor if a local AS is shown in the `AS-PATH` attribute.
- If the BGP `enforce-first-as` attribute is enabled and the update does not contain the AS number of the neighbor as the first AS number in the `AS-SEQUENCE` attribute.
- The paths which are marked as `Received-only`.

#### Support for Upstream BGP Route Export into VRFs of Peering Customers

This feature enables service providers to export or leak the upstream BGP routes into the VRFs for advertisement to peering customers. Certain peering customers usually do not require all the upstream BGP routes that the service provider receives. Instead, they want routes only from specific upstream providers. This feature gives peering customers more control over the upstream transit providers and they can choose their preferred upstream neighbor.

It allows service providers to selectively leak selected BGP routes from one instance to another without having to install the routes in the forwarding path in order to conserve the forwarding hardware resources. The leaked or exported routes do not get installed in peering customer VRFs. The feature offers an option to install the default route in the customer VRF that points to the upstream neighbor. It enables forwarding customer traffic towards the internet, connecting the customer site with external networks without requiring each specific route to be installed.

#### Policy to Notify BGP of Nexthop Changes for VPN Route Label Updates

The nexthop change via policy is notified to BGP and this will trigger a change in label. This notification enables BGP to update the labels for those routes.

### 2.3.2. BGP Configuration

#### Configuration Hierarchy

The diagram illustrates the BGP configuration hierarchy. All BGP configuration is done within an instance, for example the default instance or a VPN service instance. The instance configuration hierarchy includes parameters required for BGP but not part of the BGP configuration hierarchy itself. The BGP instance configuration hierarchy includes parameters which are generic to the respective BGP instance. The sub-hierarchies include parameters which are specific to address families, peer groups, and peers.

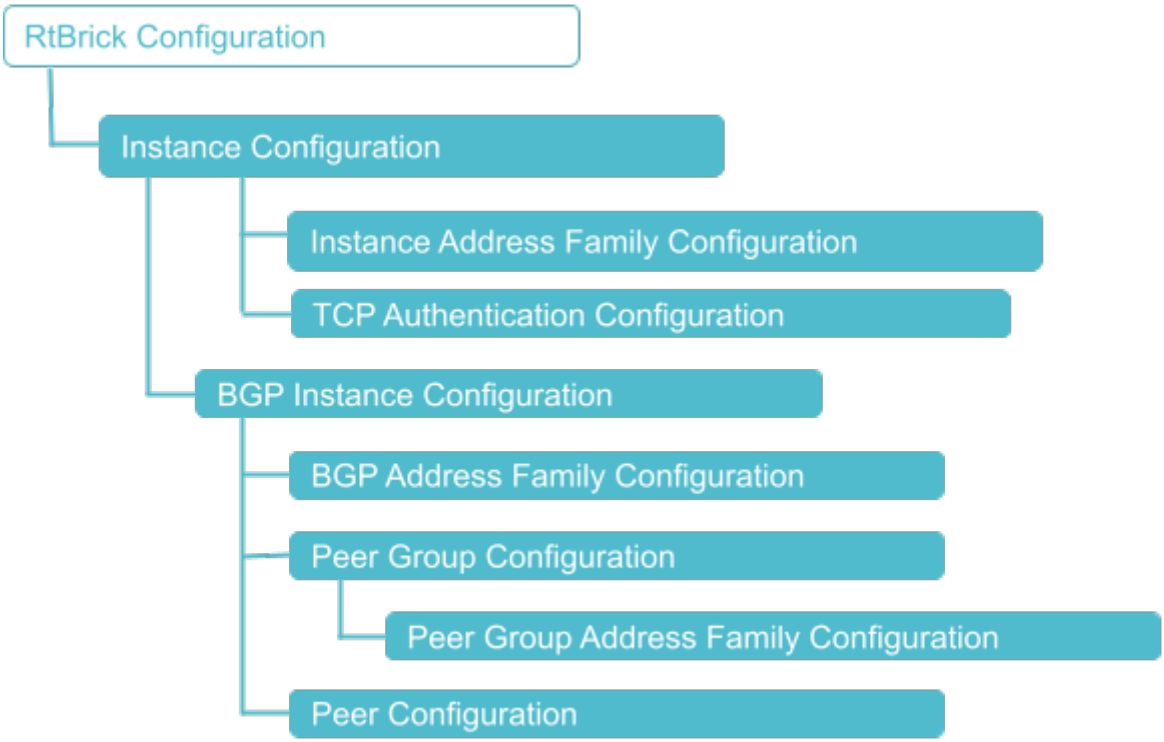



*Figure 7. BGP Configuration Hierarchy*

#### Configuration Syntax and Commands

The following sections describe the BGP configuration syntax and commands.

#### Instance Configuration

The instance configuration hierarchy includes parameters that are required for or used by BGP, but that are not part of the BGP protocol configuration hierarchy itself.

Route distinguishers and router IDs are configured directly at the instance hierarchy.

**Syntax:**

**set instance** <instance-name> <attribute> <value>

| Attribute | Description |
|---|---|
| `route-distinguisher <as-number\|ipv4-address:id>` | The route distinguisher (RD) uniquely defines routes within an IPv4 network. PE routers use route distinguishers to identify which VPN a packet belongs to. Supported formats are <as-number:id> or <ipv4-address:id>.<br>If you want to use the format <as-number:id> with a 4-byte ASN, specify it with an "L". For example, set instance services route-distinguisher 4200000000L:101 |
| `ipv4-router-id <ipv4-address>` | The router ID of the routing instance. |

The example configuration below configures the IPv4 router ID and route distinguisher for the instance `services`.

```
set instance services ipv4-router-id 198.51.100.41
set instance services route-distinguisher 198.51.100.41:101
commit
```

```
supervisor@leaf1: cfg> show config instance services
{
  "rtbrick-config:instance": {
    "name": "services",
    "ipv4-router-id": "198.51.100.41",
    "route-distinguisher": "198.51.100.41:101",
    <...>
  }
}
```

To access the RESTCONF API that corresponds to this CLI, click here.

#### Address Families

At the instance address family hierarchy, you can enable or disable address families for the instance, and configure parameters like route targets.

> The default settings depend on the instance. For the 'default' instance, the IPv4 and IPv6 unicast, multicast, and labeled unicast, as well as the MPLS unicast address families are enabled by default. For any non-default instance, no address family is enabled by default and needs to be enabled by configuration.

**Syntax:**

**set instance** <instance-name> **address-family** <afi> <safi> <attribute> <value>

| Attribute | Description |
|---|---|
| `<afi>` | Address family identifier (AFI). Supported values: ipv4, ipv6, or mpls |
| `<safi>` | Subsequent address family identifier (SAFI). Supported values: unicast, labeled-unicast, or multicast |
| `route-target ( import \| export ) <rt-value>` | Route targets (RT) are used to transfer routes between VPN instances. The RT identifies a subset of routes that should be imported to or exported from a particular VPN instance. You can configure a RT for importing or exporting routes or both.<br>If you want to use the format <as-number:id> with a 4-byte ASN, specify it with an "L". For example, set instance services address-family ipv4 unicast route-target export target:4200000000L:14 |
| `policy ( import \| export ) <policy-name>` | There are two attachment points for BGP policies. At this configuration hierarchy, you can attach import or export policies to the instance. These policies apply when routes are imported from the BGP protocol into the instance, or exported from the instance to the BGP protocol. |

The example configuration below defines policy and route targets for both IPv4 and IPv6 unicast address families within the `services` instance.

```
set instance services address-family ipv4 unicast policy export MY_V4_POLICY
set instance services address-family ipv4 unicast route-target import
target:198.51.100.70:14
set instance services address-family ipv4 unicast route-target export
target:198.51.100.70:14
set instance services address-family ipv6 unicast policy export MY_V6_POLICY
set instance services address-family ipv6 unicast route-target import
target:198.51.100.70:16
set instance services address-family ipv6 unicast route-target export
target:198.51.100.70:16
commit
```

```
supervisor@leaf1: cfg> show config instance services
{
  "rtbrick-config:instance": {
    "name": "services",
    <...>
    "address-family": [
      {
        "afi": "ipv4",
        "safi": "unicast",
        "policy": {
          "export": "MY_V4_POLICY"
        },
        "route-target": {
          "import": "target:198.51.100.70:14",
          "export": "target:198.51.100.70:14"
        }
      },
      {
        "afi": "ipv6",
        "safi": "unicast",
        "policy": {
          "export": "MY_V6_POLICY"
        },
        "route-target": {
          "import": "target:198.51.100.70:16",
          "export": "target:198.51.100.70:16"
        }
      }
    ],
    <...>
  }
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### TCP Authentication Configuration

In the instance TCP authentication hierarchy, you can optionally enable MD5 or HMAC-SHA-1-96, HMAC-SHA-256-128, or AES-128-CMAC-96 authentication. Authentication is not configured for BGP directly but for the TCP sessions used by BGP. It is necessary to bind authentication to a peer in order for the authentication to work. For details about configuring a BGP Peer, see the section Peer Configuration.

BGP TCP authentication is not backward compatible.

**Syntax:**

**set instance** <instance> **tcp authentication** <authentication-id> <attribute> <value>

| Attribute | Description |
|---|---|
| `<authentication-id>` | Authentication identifier |
| `type <type>` | Authentication identifiers such as MD5 |
| `type <type>` | Authentication identifiers such as MD5 or HMAC-SHA-1-96,HMAC-SHA-256-128 or AES-128-CMAC-96 |
| `key1-id <key1-id>` | Key ID1 of the receiver |
| `key1-encrypted-text <key1-encrypted-text>` | Encrypted text of key1 |
| `key1-plain-text <key1-plain-text>` | Plain text of key1 |
| `key2-id <key2-id>` | Key ID2 of the receiver |
| `key2-encrypted-text <key2-encrypted-text>` | Encrypted text of key2 |
| `key2-plain-text <key2-plain-text>` | Plain text of key2 |

The example below configures MD5 authentication with two keys for the `default` TCP instance.

```
set instance default tcp authentication auth1 type MD5
set instance default tcp authentication auth1 key1-id 10
set instance default tcp authentication auth1 key1-encrypted-text
"$2784cfa7523916c8cc5dfeba83562cbb4"
set instance default tcp authentication auth1 key2-id 20
set instance default tcp authentication auth1 key2-encrypted-text
"$2e9bb845e3cfcf8173973029e5c1d90d6"
commit
```

```
{
    "rtbrick-config:tcp": {
      "authentication": [
        {
          "authentication-id": "auth1",
          "type": "MD5",
          "key1-id": 10,
          "key1-encrypted-text": "$2784cfa7523916c8cc5dfeba83562cbb4",
          "key2-id": 20,
          "key2-encrypted-text": "$2e9bb845e3cfcf8173973029e5c1d90d6"
        }
      ]
    }
  }
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### BGP Instance Configuration

At this configuration hierarchy, you configure BGP protocol parameters which are generic to the BGP instance.

**Syntax:**

```
set instance <instance-name> protocol bgp <attribute> <value>
```

| Attribute | Description |
|---|---|
| `host-name <host-name>` | The name of the BGP host, to a maximum of 64 characters<br><br>` |
| domain-name <domain-name>` | The name of the BGP routing domain, to a maximum of 64 characters |
| `enforce-first-as <enable\|disable>` | By default, the BGP routing process enforces the First AS feature. It discards updates received from an eBGP peer if the peer does not list its own AS number as the first segment in the AS_PATH BGP attribute. Disable the First AS feature to accept updates without the peer's source AS matching the first AS in the AS_PATH attribute. |
| `local-as <as-number>` | The AS number in four-byte format. The numbers allowed are from 1 to 4294967295. |
| `local-preference <preference-value>` | The local preference for the BGP protocol. The numbers allowed are from 0 to 4294967295. The local preference is used to select the exit path for an AS. |
| `med <med-value>` | The BGP Multi-Exit Discriminator (MED) value. The numbers allowed are from 0 to 4294967295. When an AS has multiple links to another AS, the MED value is used to determine the exit to use to reach the other AS. |
| `protocol-preference ( internal \| external) <preference-value>` | Protocol preference of routes learned by eBGP ('external'), iBGP ('internal'), or both. This preference is used to select routes learned from multiple protocols. Lower values indicate higher preference. Range: 1 - 255. |
| `router-id <router-id>` | Router identifier in IPv4 format |
| `cluster-id <cluster-identifier>` | The cluster ID associates routers in a group within a BGP routing instance. Routers belong to the same cluster if they have the same cluster-ID. The cluster ID is formatted as an IPv4 address. |
| `timer hold-time <seconds>` | Hold timer in seconds. |
| `timer keepalive <seconds>` | Keep a live timer in seconds. |
| `bestroute-selection` <ignore-criteria> | Modify the BGP Best Path selection algorithm. For details, see the section Configuring BGP Best Path Selection. |

| Attribute | Description |
|---|---|
| `malformed-update ignore` | Ignores malformed updates, allowing the BGP session to remain active. For details, see the section [bgpug:bgp_config_cmds:::malformed-update]. |

The example below configures various global BGP parameters for the `default` instance, including cluster ID, domain name, host name, local AS, local preference, router ID, protocol preferences, and hold/keepalive timers.

```
set instance default protocol bgp cluster-id 198.51.100.51
set instance default protocol bgp domain-name rtbrick.com
set instance default protocol bgp host-name spine1
set instance default protocol bgp local-as 4200000100
set instance default protocol bgp local-preference 50
set instance default protocol bgp router-id 198.51.100.51
set instance default protocol bgp protocol-preference internal 180
set instance default protocol bgp protocol-preference external 20
set instance default protocol bgp timer hold-time 30
set instance default protocol bgp timer keepalive 10
commit
```

```
supervisor@spine1: cfg> show config instance default protocol bgp
{
  "rtbrick-config:bgp": {
    "cluster-id": "198.51.100.51",
    "domain-name": "rtbrick.com",
    "host-name": "spine1",
    "local-as": 4200000100,
    "local-preference": 50,
    "router-id": "198.51.100.51",
    "protocol-preference": {
      "internal": 180,
      "external": 20
    },
    "timer": {
      "hold-time": 30,
      "keepalive": 10
    },
    <...>
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Configuring Protocol Preference Per Address Family

RBFS supports for protocol preference configuration per address family basis in BGP.

**Syntax:**

**set instance** <instance> **protocol** <protocol> **address-family** <AFI> <SAFI> **protocol-preference [internal** | **external]** <preference-value>

| Attribute | Description |
|---|---|
| `<instance>` | Name of the routing instance. |
| `<protocol>` | Name of the protocol (e.g., bgp). |
| `<AFI>` | Address Family Identifier (e.g., ipv4, ipv6). |
| `<SAFI>` | Subsequent Address Family Identifier (e.g., unicast, labeled-unicast). |
| `[internal \| external] <preference-value>` | Preference value for internal (IBGP) and external (EBGP) routes. Lower values indicate higher preference. Range: 1 - 255. |

The example below configures the protocol preference for `external` EBGP peer routes in the BGP `labeled-unicast` address family within the `default` routing instance. The preference value indicates the priority of the routes for RIB to choose best routing among different protocols of same prefix type, with lower values being more preferred.

```
set instance default protocol bgp address-family ipv4 labeled-unicast protocol-
preference external 10
commit
```

```
supervisor@spine1: cfg> show config instance default protocol bgp address-family ipv4
labeled-unicast protocol-preference
{
  "rtbrick-config:protocol-preference": {
    "external": 10
  }
}
```

The example below configures the protocol preference for `internal` IBGP peer routes in the BGP `labeled-unicast` address family within the `default` routing instance. The preference value indicates the priority of the routes for RIB to choose best routing among different protocols of same prefix type, with lower values being more preferred.

```
set instance default protocol bgp address-family ipv4 labeled-unicast protocol-
preference internal 10
commit
```

```
supervisor@spine1: cfg> show config instance default protocol bgp address-family ipv4
labeled-unicast protocol-preference
{
  "rtbrick-config:protocol-preference": {
```

```
    "internal": 10
  }
}
```

#### Configuring BGP Best Path Selection

RBFS can be configured to ignore certain criteria when selecting the BGP best route. It is recommended to exercise extreme caution when doing this, as this can cause unexpected results such as black holes and routing loops.

**Syntax:**

**set instance** <instance-name> **protocol bgp bestroute-selection** <criteria-to-ignore>

| Criteria to ignore | Description |
|---|---|
| `ignore-as-path` | Ignore AS Path during best route selection |
| `ignore-cluster-length` | Ignore cluster-length during best route selection |
| `ignore-igp-metric` | Ignore igp metric during best route selection |
| `ignore-local-preference` | Ignore local-preference during best route selection |
| `ignore-med` | Ignore MED during best route selection |
| `ignore-origin` | Ignore origin during best route selection |
| `ignore-route-source` | Ignore route-source during best route selection |
| `ignore-validation-state` | Ignore RPKI validation state during best route selection |

The example below enables the BGP best path selection process to ignore the route source when determining the best route.

```
set instance default protocol bgp bestroute-selection ignore-route-source true
```

```
supervisor@rtbrick>LEAF01: cfg> show config instance default protocol bgp bestroute-selection
{
  "rtbrick-config:bestroute-selection": {
    "ignore-route-source": "true"
  }
}
```

#### Options for Handling Malformed BGP Update Packets

A malformed Network Layer Reachability Information (NLRI) within a BGP UPDATE message typically results in a session reset, leading to route instability and flapping. To prevent this, a new CLI option has been introduced that will ignore such malformed updates (Per RFC 7606), allowing the BGP session to remain active.

**Syntax:**

```
set instance <instance> protocol bgp malformed-update ignore
```

| Attribute | Description |
|---|---|
| <instance> | Name of the routing instance. |
| malformed-update ignore | Ignores malformed updates, allowing the BGP session to remain active. |

The following example configures a network device to ignore malformed BGP update messages for the `default` BGP instance.

```
set instance default protocol bgp malformed-update ignore
```

```
{
  "rtbrick-config:bgp": {
    "malformed-update": "ignore"
  }
}
```

#### BGP Address Family Configuration

This configuration hierarchy refers to parameters that are specific to address families but generic to the BGP instance, as opposed to peer-group specific address families configuration. At this hierarchy, you can enable or disable address families for BGP, and configure various features specific to the address family.

RBFS offers support for a configurable route resolution policy. Initially, RBFS will attempt to resolve the BGP routes using primary resolve-nexthop afi/safi configuration. In case of failure, RBFS will then proceed to resolve the BGP routes using the secondary-nexthop afi/safi configuration.

Syntax:

**set instance** <instance-name> **protocol bgp address-family** <afi> <safi> <attribute> <value>

| Attribute | Description |
|---|---|
| `<afi>` | Address family identifier (AFI). Supported values: ipv4, or ipv6 |
| `<safi>` | Subsequent address family identifier (SAFI). Supported values: unicast, labeled-unicast, vpn-unicast, multicast, or vpn-multicast |
| `default-information originate <true\|false>` | Generate and distribute a default route information |

| Attribute | Description |
|---|---|
| `download-count <count>` | Forward packets over multiple paths, set maximum prefixes to use |
| `multipath <number>` | Enable load sharing among multiple BGP paths<br>The default ECMP Maximum Next-hop Count is limited to 8; however, this limit varies depending on the platform that RBFS supports. For details on the limits set for the ECMP Maximum Next-hop Count on the platforms, refer to the Resource Limits/Feature Support section of the *Platform Guide*. |
| `retain-route-target (enable\|disable)` | Retain VPN routes for all route targets, by default this feature is enabled |
| `resolve-nexthop afi <afi>` | Address family to resolve the next-hop |
| `resolve-nexthop safi <safi>` | Sub-address family to resolve the next-hop |
| `resolve-nexthop-secondary afi <afi>` | Address family to resolve the next-hop when the primary route resolution fails |
| `resolve-nexthop-secondary safi <safi>` | Sub-address family to resolve the next-hop when the primary route resolution fails |
| `redistribute <source>` | Enable the redistribution feature to dynamically inject specific types of routes into the BGP protocol. Supported route sources are `direct`, `igmp`, `ipoe`, `isis`, `ospf`, `pim`, `ppp`, `static`, and `arp-nd`. |
| `redistribute <source> policy <policy>` | Attach a policy to the redistribution process |
| `srgb base <value>` | Segment Routing Global Block (SRGB) start label. The SRGB is the range of label values reserved for segment routing (SR). These values are assigned as segment identifiers (SIDs) to SR-enabled network nodes and have global significance throughout the routing domain. SRGB is supported for labeled unicast only. |
| `srgb index <value>` | Segment Routing Global Block (SRGB) index |
| `srgb range <value>` | Segment Routing Global Block (SRGB) label range<br>For details on the supported label ranges, see the Static, System, and Dynamic Label Ranges section in the Platform Guide. |
| inter-as-option-ab enable | Enables Inter-AS Option AB for the specified address family. For details, see the Inter-AS Option A-B Configuration section of this guide. |

The example configuration below enables various IPv4 and IPv6 address families for the `default` BGP instance, including VPN unicast and labeled unicast with specific SRGB parameters.

```
set instance default protocol bgp address-family ipv4 vpn-unicast
set instance default protocol bgp address-family ipv6 labeled-unicast srgb base 5000
set instance default protocol bgp address-family ipv6 labeled-unicast srgb range 1000
set instance default protocol bgp address-family ipv6 labeled-unicast srgb index 11
set instance default protocol bgp address-family ipv6 unicast
set instance default protocol bgp address-family ipv6 vpn-unicast
commit
```

```
supervisor@spine1: cfg> show config instance default protocol bgp
{
  "rtbrick-config:bgp": {
    <...>
    "address-family": [
      {
        "afi": "ipv4",
        "safi": "vpn-unicast"
      },
      {
        "afi": "ipv6",
        "safi": "labeled-unicast",
        "srgb": {
          "base": 5000,
          "range": 1000,
          "index": 11
        }
      },
      {
        "afi": "ipv6",
        "safi": "unicast"
      },
      {
        "afi": "ipv6",
        "safi": "vpn-unicast"
      }
    ],
    <...>
  }
}
```

The example configuration below redistributes direct, PPP, ARP-ND, and static routes into BGP for both IPv4 and IPv6 unicast address families within the `services` instance.

```
set instance services protocol bgp address-family ipv4 unicast redistribute direct
set instance services protocol bgp address-family ipv4 unicast redistribute ppp
set instance services protocol bgp address-family ipv4 unicast redistribute arp-nd
set instance services protocol bgp address-family ipv4 unicast redistribute static
set instance services protocol bgp address-family ipv6 unicast redistribute direct
set instance services protocol bgp address-family ipv6 unicast redistribute ppp
```

```
set instance services protocol bgp address-family ipv6 unicast redistribute static
commit
```

```
supervisor@leaf1: cfg> show config instance services protocol bgp
{
  "rtbrick-config:bgp": {
    <...>
    "address-family": [
      {
        "afi": "ipv4",
        "safi": "unicast",
        "redistribute": [
          {
            "source": "direct"
          },
          {
            "source": "ppp"
          },
          {
            "source": "arp-nd"
          },
          {
            "source": "static"
          }
        ]
      },
      {
        "afi": "ipv6",
        "safi": "unicast",
        "redistribute": [
          {
            "source": "direct"
          },
          {
            "source": "ppp"
          },
          {
            "source": "static"
          }
        ]
      }
    ]
  }
}
```

The example configuration below shows how to redistribute direct, PPP, and static routes into BGP for both IPv4 and IPv6 unicast address families within a `services` instance.

```
set instance services protocol bgp address-family ipv4 unicast redistribute direct
policy MY_REDISTRIBUTION_POLICY
set instance services protocol bgp address-family ipv4 unicast redistribute ppp
set instance services protocol bgp address-family ipv4 unicast redistribute static
set instance services protocol bgp address-family ipv6 unicast redistribute direct
policy MY_REDISTRIBUTION_POLICY
set instance services protocol bgp address-family ipv6 unicast redistribute ppp
set instance services protocol bgp address-family ipv6 unicast redistribute static
commit
```

```
supervisor@leaf1: cfg> show config instance services protocol bgp
{
  "rtbrick-config:bgp": {
    <...>
    "address-family": [
      {
        "afi": "ipv4",
        "safi": "unicast",
        "redistribute": [
          {
            "source": "direct"
            "policy": "MY_REDISTRIBUTION_POLICY"
          },
          {
            "source": "ppp"
          },
          {
            "source": "static"
          }
        ]
      },
      {
        "afi": "ipv6",
        "safi": "unicast",
        "redistribute": [
          {
            "source": "direct"
            "policy": "MY_REDISTRIBUTION_POLICY"
          },
          {
            "source": "ppp"
          },
          {
            "source": "static"
          }
        ]
      }
    ]
  }
```

```
}
```

In the example below, RBFS would resolve the BGP routes in the IPv4 labeled-unicast RIB. If this fails, then the routes would be resolved in IPv4 unicast RIB.

```
set instance default protocol bgp address-family ipv4 unicast resolve-nexthop afi ipv4
set instance default protocol bgp address-family ipv4 unicast resolve-nexthop safi
labeled-unicast
set instance default protocol bgp address-family ipv4 unicast resolve-nexthop-
secondary afi ipv4
set instance default protocol bgp address-family ipv4 unicast resolve-nexthop-
secondary safi unicast
commit
```

```
supervisor@leaf1: cfg> show config instance default protocol bgp address-family ipv4
unicast
{
  "rtbrick-config:address-family": [
    {
      "afi": "ipv4",
      "safi": "unicast",
      "resolve-nexthop": {
        "afi": "ipv4",
        "safi": "labeled-unicast"
      },
      "resolve-nexthop-secondary": {
        "afi": "ipv4",
        "safi": "unicast"
      }
    }
  ]
}
```

The example below configures the BGP protocol for IPv4 unicast within the default instance. It then enables the redistribution of ARP and ND (Neighbor Discovery) entries into BGP, applying a policy named ARP_ND_SOURCE_MATCH to control which entries are redistributed.

```
set instance default protocol bgp address-family ipv4 unicast
set instance default protocol bgp address-family ipv4 unicast redistribute arp-nd
set instance default protocol bgp address-family ipv4 unicast redistribute arp-nd
policy ARP_ND_SOURCE_MATCH
```

```
{
  "rtbrick-config:address-family": [
    {
      "afi": "ipv4",
```

```
      "safi": "unicast",
      "redistribute": [
        {
          "source": "arp-nd",
          "policy": [
            "ARP_ND_SOURCE_MATCH"
            ]
        }
      ]
    }
  ]
}
```

For details about matching ARP-ND source in policy, see the Matching ARP-ND Source in Policy section of the *Policy User Guide*.

To access the RESTCONF API that corresponds to this CLI, click here.

#### Peer Group Configuration

##### Peer Groups

In BGP, neighbor peers with the same update policies can be grouped to simplify the initial configuration and updates. Peers share the same policies such as route maps, distribution lists, filter lists, update sources, and so on, so peer groups only need one configuration statement for these values.

**Syntax:**

**set instance** <instance-name> **protocol bgp peer-group** <peer-group-name> <attribute> <value>

| Attribute | Description |
|---|---|
| `local-as <as-number>` | Local AS number for the peer group |
| `remote-as <as-number>` | Remote AS number for the peer group |
| `any-as <true\|false>` | Enable dynamic AS negotiation for this peer group |
| `ebgp-multihop <hop-count>` | By default, the maximum number of hops between eBGP peers is 1 (direct connection). This hop count overrides the default behavior allowing connectivity between eBGP peers not directly connected. |
| `link-local-nexthop-only <true\|false>` | Enable BGPv6 peerings using the IPv6 link-local addresses |
| `no-prepend <true\|false>` | Do not prepend the local AS for advertisements to the peer |
| `replace-as <true\|false>` | Prepend only the local AS for advertisements to the peer |
| `ttl-security <enable\|disable>` | Enables or disalbes Generalized TTL Security Mechanism (GTSM). |

| Attribute | Description |
|---|---|
| `ttl-limit <ttl-limit>` | Specifies the minimum TTL value of packets from the BGP neighbor for TTL Security. The valid range is 1 to 255. |

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Address Families

At this configuration hierarchy, you can enable the address families that shall be supported for the group peers, and enable features specific to the address family. By default, BGP neighbor sessions support the IP4v unicast and multicast address families.

**Syntax:**

> **set instance** <instance-name> **protocol bgp peer-group** <peer-group-name> **address-family** <afi> <safi> <attribute> <value>

| Attribute | Description |
|---|---|
| `<afi>` | Address family identifier (AFI). Supported values: ipv4, or ipv6 |
| `<safi>` | Subsequent address family identifier (SAFI). Supported values: unicast, labeled-unicast, vpn-unicast, multicast, or vpn-multicast |
| `add-path` | Negotiate additional path capabilities with these peers, so that more than one path can be active to the peers in the group |
| `default-information originate <true\|false>` | Generate and advertise a default route to peers in the group |
| `extended-nexthop` | Enable extended-next-hop encoding for BGP peer groups to allow the transfer of IPv4 prefixes over an IPv6 connection |
| `nexthop-self <true\|false>` | Set the advertised BGP nexthop to yourself, this is the default for eBGP |
| `nexthop-unchanged <true\|false>` | Do not modify the advertised BGP nexthop, this is the default for iBGP |
| `update-nexthop ( ipv4-address \| ipv6-address )` <address> | BGP nexthop address for routes advertised to this peer group |
| `remove-private-as <true\|false>` | Remove private AS numbers from routes advertised to group peers |
| `route-reflect-client <true\|false>` | Configure this peer as a route reflector client |
| `policy ( import \| export ) <policy-name>` | Apply a routing policy to the peer group |

The example configuration below sets up a BGP peer-group named "spine" with specific parameters for IPv4 and IPv6 address families, including the use of link-local nexthops, a remote AS number, and extended nexthop and update nexthop settings for VPN unicast routes.

```
set instance default protocol bgp peer-group spine link-local-nexthop-only true
set instance default protocol bgp peer-group spine remote-as 4200000100
set instance default protocol bgp peer-group spine address-family ipv4 vpn-unicast
extended-nexthop true
set instance default protocol bgp peer-group spine address-family ipv4 vpn-unicast
update-nexthop ipv6-address 2001:db8:0:19::
set instance default protocol bgp peer-group spine address-family ipv6 labeled-unicast
set instance default protocol bgp peer-group spine address-family ipv6 unicast
set instance default protocol bgp peer-group spine address-family ipv6 vpn-unicast
update-nexthop ipv6-address 2001:db8:0:19::
commit
```

```
supervisor@leaf1: cfg> show config instance default protocol bgp peer-group spine
{
  "rtbrick-config:peer-group": {
    "pg-name": "spine",
    "link-local-nexthop-only": "true",
    "remote-as": 4200000100,
    "address-family": [
      {
        "afi": "ipv4",
        "safi": "vpn-unicast",
        "extended-nexthop": "true",
        "update-nexthop": {
          "ipv6-address": "2001:db8:0:19::"
        }
      },
      {
        "afi": "ipv6",
        "safi": "labeled-unicast"
      },
      {
        "afi": "ipv6",
        "safi": "unicast"
      },
      {
        "afi": "ipv6",
        "safi": "vpn-unicast",
        "update-nexthop": {
          "ipv6-address": "2001:db8:0:19::"
        }
      }
    ]
  }
}
```

The example configuration below defines a BGP peer-group named "ipv4_bgp" with a remote AS, enables TTL security with a limit of 253, and activates both IPv4 and IPv6 unicast address families

for this peer-group.

```
set instance default protocol bgp peer-group ipv4_bgp remote-as 4200000100
set instance default protocol bgp peer-group ipv4_bgp ttl-security enable
set instance default protocol bgp peer-group ipv4_bgp ttl-limit 253
set instance default protocol bgp peer-group ipv4_bgp address-family ipv4 unicast
set instance default protocol bgp peer-group ipv4_bgp address-family ipv6 unicast
commit
```

```
{
  "rtbrick-config:peer-group": [
    {
      "pg-name": "ipv4_bgp",
      "remote-as": 4200000100,
      "ttl-security": "enable",
      "ttl-limit": 253,
      "address-family": [
        {
          "afi": "ipv4",
          "safi": "unicast"
        },
        {
          "afi": "ipv6",
          "safi": "unicast"
        }
      ]
    }
  ]
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Maximum Prefix Limit

The BGP Maximum Prefix Limit feature enables you to set a limit for the maximum number of prefixes that a BGP router can receive from its peer router. If a BGP router receives prefixes that exceed the defined limit threshold, the BGP session gets reset and the session goes idle for a pre-defined period.

You can define a period as idle timeout so that the BGP peering gets re-established automatically after the specified time. If you do not specify the idle timeout, the BGP peering does not get re-established until or unless you execute the `clear bgp peer` command.

Before getting into inactive or idle mode, the router sends a notification message to the peer router about the exceeded threshold with the error code and the sub-code.

You can configure prefix limits for a peer group.

**Syntax:**

**set instance** <instance-name> **protocol bgp peer-group** <peer-group-name> **address-family** <afi> <safi> **prefix-limit** <attribute> <value>

| Attribute | Description |
|---|---|
| `<afi>` | Address family identifier (AFI). Supported values: ipv4, or ipv6 |
| `<safi>` | Subsequent address family identifier (SAFI). Supported values: unicast, labeled-unicast, vpn-unicast, or vpn-multicast |
| `count <count>` | Number of maximum prefixes that the peer router is allowed to send. The default value is 0. It means no value is configured for prefix limit. |
| `idle-timeout <idle-timeout>` | Idle or inactive time after the maximum limit is reached (in minutes). The allowed range is 1 - 2400 min. The default is 'Forever'. |

The example configuration below configures a prefix limit of 100 with an idle timeout of 5 seconds for the IPv4 unicast address family within the BGP peer-group "PE2" in the default instance.

```
set instance default protocol bgp peer-group PE2 address-family ipv4 unicast prefix-
limit count 100
set instance default protocol bgp peer-group PE2 address-family ipv4 unicast prefix-
limit idle-timeout 5
commit
```

```
supervisor@L1-STD-2-2002>bm14-tst.fsn.rtbrick.net: cfg> show config instance default
protocol bgp peer-group PE2 address-family ipv4 unicast
{
  "rtbrick-config:address-family": [
    {
      "afi": "ipv4",
      "safi": "unicast",
      "prefix-limit": {
        "count": 100,
        "idle-timeout": 5
      }
    }
  ]
}
```

To access the RESTCONF API that corresponds to this CLI, click here.

#### Configure Upstream BGP Route Export into VRFs of Peering Customers

For peering customers, it is required to select and install specific upstream routes in their own dedicated routing tables (VRFs). To achieve this, the routes from a single upstream peer must be

distributed to multiple RIBs, one for each customer-specific VRF.

You can configure the Upstream BGP Route Export into VRFs of peering customers.

**Syntax:**

**set instance** <instance-name> **protocol bgp peer-group** <peer-group-name> **address-family** <afi> <safi> **export-rib** <name>

| Attribute | Description |
|---|---|
| `<instance-name>` | Name of the routing instance. |
| `<peer-group-name>` | Name of the peer group. |
| `<afi>` | Address family identifier (AFI). Supported values: ipv4, or ipv6 |
| `<safi>` | Subsequent address family identifier (SAFI). Supported values: unicast, labeled-unicast, vpn-unicast, or vpn-multicast |
| `export-rib` | Defines in which routing tables (RIBs) the routes from this peer will be installed. The option indicates that the routes learned from the BGP peer are to be installed in multiple RIBs. |
| `peer-default-route` | It specifies the default route that is to be installed in the customer's routing table. |

The example configuration below configures default route advertisement and export of RIBs for IPv4 unicast within the BGP peer-group "PE2" to peers CE1, CE2, and CE3 in the `default` instance.

```
set instance default protocol bgp peer-group PE2 address-family ipv4 unicast peer-
default-route CE1
set instance default protocol bgp peer-group PE2 address-family ipv4 unicast peer-
default-route CE2
set instance default protocol bgp peer-group PE2 address-family ipv4 unicast peer-
default-route CE3
set instance default protocol bgp peer-group PE2 address-family ipv4 unicast export-
rib CE1
set instance default protocol bgp peer-group PE2 address-family ipv4 unicast export-
rib CE2
set instance default protocol bgp peer-group PE2 address-family ipv4 unicast export-
rib CE3
commit
```

```
supervisor@rtbrick.net: cfg> show config instance default protocol bgp peer-group PE2
address-family ipv4 unicast
{
  "rtbrick-config:address-family": [
    {
      "afi": "ipv4",
```

```
      "safi": "unicast",
      "peer-default-route": [
        "CE1",
        "CE2",
        "CE3"
        ],
      "export-rib": [
        {
          "instance": "CE1"
        },
        {
          "instance": "CE2"
        },
        {
          "instance": "CE3"
        }
      ]
    }
  ]
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Peer Configuration

Once peer groups have been defined, BGP peers can be configured at the peer configuration hierarchy. A peer can be specified by address, or by interface when using IPv6 auto-discovered neighbors and link-local addresses. Furthermore, it is possible to configure TCP authentication and bind it to a peer.

Syntax to configure a BGP peer by address:

**set instance** <instance-name> **protocol bgp peer** ( ipv4 | ipv6) <peer-address> <update-source> **peer-group** <peer-group>

Syntax to configure a BGP peer using IPv6 link-local addresses:

**set instance** <instance-name> **protocol bgp peer interface** <name> **peer-group** <peer-group>

Syntax to configure TCP Authentication for BGP peers:

**set instance** <instance-name> **protocol bgp peer** (ipv4 | ipv6) <peer-address> <update-source> **authentication-id** <authentication-id>

| Attribute | Description |
|---|---|
| `interface <name>` | Enable BGP peer using IPv6 link-local addresses |
| `ipv4 <peer-address>` | IPv4 address of a BGP peer |
| `ipv6 <peer-address>` | IPv6 address of a BGP peer |
| `allow-as-in <value>` | Specify the value for allow-as-in. Allowed range of value 1 - 10. |
| `<update-source>` | Local IP address to be used for the peering |
| `peer-group <peer-group>` | Assign the peer to a peer group |
| `deactivate` | Deactivate a configured peer |
| `authentication-id <authentication-id>` | Authentication identifier |

The example configuration below configures update source for a BGP peer to a specific IP address and assigns the peer to the "spine" peer-group within the `default` instance.

```
set instance default protocol bgp peer ipv4 peer-address 198.51.100.82 update-source
198.51.100.81
set instance default protocol bgp peer ipv4 peer-address 198.51.100.82 peer-group
spine
commit
```

```
supervisor@rtbrick: cfg> show config instance default protocol bgp peer

{
  "rtbrick-config:peer": {
    "ipv4": [
      {
        "peer-address": "198.51.100.82",
        "update-source": "198.51.100.81",
        "peer-group": "spine"
      }
    ]
  }
}
```

The example configuration below assigns a specific interface to the "spine" peer-group within the BGP protocol configuration of the `default` instance.

```
set instance default protocol bgp peer interface ifl-0/0/1/1 peer-group spine
commit
```

```
supervisor@rtbrick: cfg> show config instance default protocol bgp peer
{
  "rtbrick-config:peer": {
```

```
    "interface": [
      {
        "name": "ifl-0/0/1/1",
        "peer-group": "spine"
      }
    ]
  }
}
```

The example configuration below applies an authentication ID to a BGP peer interface and assigns that interface to the "spine" peer-group within the default instance.

```
set instance default protocol bgp peer interface ifl-0/0/1/1 authentication-id auth1
set instance default protocol bgp peer interface ifl-0/0/1/1 peer-group spine
commit
```

```
supervisor@rtbrick: cfg> show config instance default protocol bgp peer
{
  "rtbrick-config:peer": {
    "interface": [
      {
        "name": "ifl-0/0/1/1",
        "authentication-id": "auth1",
        "peer-group": "spine"
      }
    ]
  }
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

### Inter-AS Option A-B Configuration

The VPN Option A-B Interconnect solution is aimed at conserving label space in spine devices by using a single static label for multiple VPN prefixes.

#### Option-B Label Space Conservation

The MSANs are connected via an L3VPN private network. Traditionally, spines allocate a new service label (Option B) for each next hop and label when advertising VPN prefixes to the Route Reflector (RR) in the IP2 core. This can lead to a large number of MPLS Option B labels being installed in the spines. This solution addresses this by enabling a more efficient label allocation.

To conserve label space, a new MSAN POTS instance is configured on the spines with a static label. All MSAN routes are installed in this instance, and their service labels are replaced with the single static label before being forwarded to the RR. When data packets arrive from the core, an IP lookup followed by an MPLS push is performed, and packets are forwarded towards the leaf using this single label.

The key features of this solution are:

- Supports the static label for an address family (IPv4 unicast / IPv6 unicast).
- Supports enabling/disabling of VPN address family (IPv4 vpn-unicast / IPv6 vpn-unicast).
- Supports replacing the service label with the configured static instance label if the prefix RT matches the import RT of the instance.

#### Configuring Inter-AS Option A-B

The `inter-as-option-ab` parameter needs to be enabled within the BGP configuration for vpn-unicast address families (both IPv4 and IPv6) under the relevant instance.

**Syntax**:

**set instance** <instance-name> **protocol bgp address-family** <afi> <safi> <attribute> <value>

| Attribute | Description |
|---|---|
| <afi> | Address family identifier (AFI) for the address family (`IPv4` / `IPv6`) |
| <safi> | Subsequent address family identifier (SAFI) for the address family (`vpn-unicast`) |
| inter-as-option-ab enable | Enables Inter-AS Option AB for the specified address family. |

The following example enables Inter-AS Option AB for both IPv4 and IPv6 VPN-unicast address families under the `default` instance:

```
set instance default protocol bgp local-as 65001
set instance default protocol bgp address-family ipv4 unicast
set instance default protocol bgp address-family ipv4 vpn-unicast inter-as-option-ab
enable
set instance default protocol bgp address-family ipv6 vpn-unicast inter-as-option-ab
enable
commit
```

**Configuration Example**:

```
"rtbrick-config:instance": [
  {
    "name": "default",
    "protocol": {
      "bgp": {
        "local-as": 65001,
        "address-family": [
          {
            "afi": "ipv4",
            "safi": "unicast"
```

```
          },
          {
            "afi": "ipv4",
            "safi": "vpn-unicast",
            "inter-as-option-ab": "enable"
          },
          {
            "afi": "ipv6",
            "safi": "vpn-unicast",
            "inter-as-option-ab": "enable"
          }
        ]
      }
    }
  }
]
```

#### Configuring Static Label

**Syntax**:

**set instance** <instance-name> **protocol bgp address-family** <afi> <safi> <attribute> <value>

| Attribute | Description |
|---|---|
| <afi> | Address family identifier (AFI) for the address family (`IPv4` / `IPv6`) |
| <safi> | Subsequent address family identifier (SAFI) for the address family (`unicast`) |
| <afi> <safi> label <label> | Static label for IPv4/IPv6 unicast address family. |

The following example enables static label for both IPv4 and IPv6 unicast address families under the `vrf1` instance:

```
set instance vrf1 protocol bgp address-family ipv4 unicast
set instance vrf1 protocol bgp address-family ipv4 unicast label 400000
set instance vrf1 protocol bgp address-family ipv6 unicast
set instance vrf1 protocol bgp address-family ipv6 unicast label 401000
commit
```

**Configuration Example**:

```
{
  "rtbrick-config:address-family": [
    {
      "afi": "ipv4",
      "safi": "unicast",
```

```
      "label": 400000
    },
    {
      "afi": "ipv6",
      "safi": "unicast",
      "label": 401000
    }
  ]
}
```

**Multiple MPLS Label Support for BGP Labeled Unicast**

RBFS support for receiving and sending multiple MPLS labels that enable BGP Label Unicast (BGP-LU) for executing traffic engineering policies (as per RFC8277). RBFS allows the use of up to eight MPLS labels on Q2C and Q2A platforms. You can enable 'multi-label capability' option for both IPv4 and IPv6 labeled-unicast address families to accommodate advanced traffic engineering use cases.

This feature enhances that capability by enabling BGP to advertise multiple MPLS labels stacked together for the same Network Layer Reachability Information (NLRI) or prefix. This capability allows BGP-Labeled Unicast (BGP-LU) to provide detailed forwarding information and to contain complex transport and service topologies. Also, this feature simplifies operations, particularly when controllers manage label stacks centrally.

It allows RBFS routers to receive, process, and advertise BGP LU routes that contain a multiple MPLS labels. It enables the execution of controller-driven traffic engineering policies. Controllers can decide optimized paths using Segment Routing parameters and encode these paths as MPLS label stacks. The RBFS capability to install and push multiple labels on outgoing packets is used to forward traffic along these controller-defined paths.

This feature allows the controller to define optimized paths using Segment Routing, encode them as MPLS label stacks, and advertise them using BGP LU for IPv4 and IPv6. RBFS installs and pushes these labels on outgoing packets to forward traffic along controller-defined paths.

**Syntax:**

**set instance** <instance-name> **protocol bgp peer-group** <peer-group> **address-family [ipv4 | ipv6] labeled-unicast multi-label-capability true**

| Attribute | Description |
|---|---|
| `<instance-name>` | Name of the instance. |
| `<afi>` | Address family identifier (AFI). Supported values: ipv4, or ipv6 |
| `<safi>` | Subsequent address family identifier (SAFI). Supported labeled-unicast. |
| `multi-label capability true` | Set this to true. It enables both IPv4 and IPv6 labeled-unicast address families to execute advanced traffic engineering policies. |

The following configuration enables multi-label-capability under IPv4 and IPv6 labeled-unicast. It allows the BGP peer-group PE2_SDN to negotiate and exchange routes that carry multiple MPLS

labels. This applies to both IPv4 and IPv6 address families with SAFI labeled-unicast. This configuration , ensures that the peer can advertise and receive NLRI with a label stack instead of a single label.

Example for IPv4:

```
set instance default protocol bgp peer-group PE2_SDN address-family ipv4 labeled-
unicast multi-label-capability true
```

```
supervisor@rtbrick.net: cfg> show config instance default protocol bgp peer-group
PE2_SDN address-family ipv4 labeled-unicast
{
  "rtbrick-config:address-family": [
    {
      "afi": "ipv4",
      "safi": "labeled-unicast",
      "multi-label-capability": "true"
    }
  ]
}
```

Example for IPv6:

```
set instance default protocol bgp peer-group PE2_SDN address-family ipv6 labeled-
unicast multi-label-capability true
```

```
supervisor@rtbrick.net: cfg> show config instance default protocol bgp peer-group
PE2_SDN address-family ipv6 labeled-unicast
{
  "rtbrick-config:address-family": [
    {
      "afi": "ipv6",
      "safi": "labeled-unicast",
      "multi-label-capability": "true"
    }
  ]
}
```

**BGP Sample Configuration**

Example 1: BGP Configuration of a Spine Switch (Default Instance only)

```
{
  "ietf-restconf:data": {
    "rtbrick-config:instance": [
```

```
      {
        "name": "default",
        "ipv4-router-id": "198.51.100.51",
        "protocol": {
          "bgp": {
            "domain-name": "rtbrick.com",
            "host-name": "spine1",
            "local-as": 4200000100,
            "address-family": [
              {
                "afi": "ipv4",
                "safi": "vpn-unicast"
              },
              {
                "afi": "ipv6",
                "safi": "labeled-unicast",
                "srgb": {
                  "base": 5000,
                  "range": 1000,
                  "index": 11
                },
                "redistribute": [
                  {
                    "source": "direct"
                  }
                ]
              },
              {
                "afi": "ipv6",
                "safi": "unicast",
                "redistribute": [
                  {
                    "source": "direct"
                  }
                ]
              },
              {
                "afi": "ipv6",
                "safi": "vpn-unicast"
              }
            ],
            "peer": {
              "interface": [
                {
                  "name": "ifl-0/1/1/1",
                  "authentication-id": "auth1",
                  "peer-group": "spine"
                },
                {
                  "name": "ifl-0/2/1/1",
                  "peer-group": "leaf1"
```

```
                },
                {
                  "name": "ifl-0/2/2/1",
                  "peer-group": "leaf2"
                }
              ]
            },
            "peer-group": [
              {
                "pg-name": "leaf1",
                "link-local-nexthop-only": "true",
                "remote-as": 4200000201,
                "address-family": [
                  {
                    "afi": "ipv4",
                    "safi": "vpn-unicast",
                    "extended-nexthop": "true",
                    "nexthop-unchanged": "true"
                  },
                  {
                    "afi": "ipv6",
                    "safi": "labeled-unicast"
                  },
                  {
                    "afi": "ipv6",
                    "safi": "unicast"
                  },
                  {
                    "afi": "ipv6",
                    "safi": "vpn-unicast",
                    "nexthop-unchanged": "true"
                  }
                ]
              },
              {
                "pg-name": "leaf2",
                "link-local-nexthop-only": "true",
                "remote-as": 4200000202,
                "address-family": [
                  {
                    "afi": "ipv4",
                    "safi": "vpn-unicast",
                    "extended-nexthop": "true",
                    "nexthop-unchanged": "true"
                  },
                  {
                    "afi": "ipv6",
                    "safi": "labeled-unicast"
                  },
                  {
                    "afi": "ipv6",
```

```
                "safi": "unicast"
              },
              {
                "afi": "ipv6",
                "safi": "vpn-unicast",
                "nexthop-unchanged": "true"
              }
            ]
          },
          {
            "pg-name": "spine",
            "link-local-nexthop-only": "true",
            "remote-as": 4200000100,
            "address-family": [
              {
                "afi": "ipv4",
                "safi": "vpn-unicast",
                "extended-nexthop": "true"
              },
              {
                "afi": "ipv6",
                "safi": "labeled-unicast",
                "nexthop-self": "true"
              },
              {
                "afi": "ipv6",
                "safi": "unicast",
                "nexthop-self": "true"
              },
              {
                "afi": "ipv6",
                "safi": "vpn-unicast"
              }
            ]
          }
        ]
      }
    }
  }
  ]
}
}
```

Example 2: BGP Configuration of a Leaf Switch with one VPN Instance

```
{
  "ietf-restconf:data": {
    "rtbrick-config:instance": [
      {
        "name": "default",
```

```
        "ipv4-router-id": "198.51.100.53",
        "protocol": {
          "bgp": {
            "domain-name": "rtbrick.com",
            "host-name": "leaf1",
            "local-as": 4200000201,
            "address-family": [
              {
                "afi": "ipv4",
                "safi": "vpn-unicast"
              },
              {
                "afi": "ipv6",
                "safi": "labeled-unicast",
                "srgb": {
                  "base": 5000,
                  "range": 1000,
                  "index": 13
                },
                "redistribute": [
                  {
                    "source": "direct"
                  }
                ]
              },
              {
                "afi": "ipv6",
                "safi": "unicast",
                "redistribute": [
                  {
                    "source": "direct"
                  }
                ]
              },
              {
                "afi": "ipv6",
                "safi": "vpn-unicast"
              }
            ],
            "peer": {
              "interface": [
                {
                  "name": "ifl-0/1/1/1"
                  "authentication-id": "auth1",
                  "peer-group": "spine"
                },
                {
                  "name": "ifl-0/1/2/1",
                  "peer-group": "spine"
                }
              ]
```

```
            },
            "peer-group": [
              {
                "pg-name": "spine",
                "link-local-nexthop-only": "true",
                "remote-as": 4200000100,
                "address-family": [
                  {
                    "afi": "ipv4",
                    "safi": "vpn-unicast",
                    "extended-nexthop": "true",
                    "update-nexthop": {
                      "ipv6-address": "2001:db8:0:19::"
                    }
                  },
                  {
                    "afi": "ipv6",
                    "safi": "labeled-unicast"
                  },
                  {
                    "afi": "ipv6",
                    "safi": "unicast"
                  },
                  {
                    "afi": "ipv6",
                    "safi": "vpn-unicast",
                    "update-nexthop": {
                      "ipv6-address": "2001:db8:0:19::"
                    }
                  }
                ]
              }
            ]
          }
        }
      },
      {
        "name": "services",
        "ipv4-router-id": "198.51.100.41",
        "route-distinguisher": "198.51.100.41:101",
        "address-family": [
          {
            "afi": "ipv4",
            "safi": "unicast",
            "policy": {
              "export": "MY_V4_POLICY"
            },
            "route-target": {
              "import": "target:198.51.100.70:14",
              "export": "target:198.51.100.70:14"
            }
```

```
          },
          {
            "afi": "ipv6",
            "safi": "unicast",
            "policy": {
              "export": "MY_V6_POLICY"
            },
            "route-target": {
              "import": "target:198.51.100.70:16",
              "export": "target:198.51.100.70:16"
            }
          }
        ],
        "protocol": {
          "bgp": {
            "domain-name": "rtbrick.com",
            "host-name": "leaf1",
            "local-as": 65003,
            "address-family": [
              {
                "afi": "ipv4",
                "safi": "unicast",
                "redistribute": [
                  {
                    "source": "direct"
                  },
                  {
                    "source": "ppp"
                  },
                  {
                    "source": "static"
                  }
                ]
              },
              {
                "afi": "ipv6",
                "safi": "unicast",
                "redistribute": [
                  {
                    "source": "direct"
                  },
                  {
                    "source": "ppp"
                  },
                  {
                    "source": "static"
                  }
                ]
              }
            ]
          }
```

```
        }
      }
    ]
  }
}
```

### 2.3.3. BGP Operational Commands

#### BGP Show Commands

The BGP show commands provide detailed information about the BGP protocol operation and BGP routes.

##### BGP Summary

This command displays BGP protocol parameters like attributes or timers that are generic to the BGP instance.

**Syntax:**

**show bgp summary** <option>

| Option | Description |
|---|---|
| - | Without any option, the commands displays the information for all instances. |
| instance <instance-name> | BGP summary information for the given instance. |

Example: BGP summary for the default instance

```
supervisor@rtbrick: op> show bgp summary instance default
Instance: default
  General information
    Hostname: PE1, Domain name: Rtbrick
    Local AS: 65541, Version: 4
    Local preference: 100, eBGP Protocol preference: 20, iBGP Protocol preference: 200
    Router ID: 198.51.100.11, Cluster ID: 198.51.100.11
    MED: 0, Enforce first AS: True
    Address families configured:
      ipv4: labeled-unicast, unicast
      ipv6: labeled-unicast, unicast
  Capabilities
    Route refresh: True, AS4: True, Graceful restart: False,
  Best route selection
    Always compare MED: True, Ignore as path: False
    Ignore local preference: False, Ignore origin: False
    Ignore MED: False, Ignore route source: False
    Ignore router ID: False, Ignore uptime: True
```

```
    Ignore cluster length: False, Ignore peer IP: False
    Route select parameter: 0
  Timers
    Connect retry: 30s, Keepalive: 30s, Holdtime: 90s
  Statistics
    Peers configured: 1, Peers auto discovery: 0
      Peers in idle          : 0
      Peers in connect       : 0
      Peers in active        : 0
      Peers in opensent      : 0
      Peers in openconfirm   : 0
      Peers in established   : 1
```

API To access the Operational State API that corresponds to this CLI, click here.

#### BGP Peer

The 'show bgp peer' commands display information on BGP peers.

**Syntax:**

**show bgp peer** <option> ...

| Option | Description |
|---|---|
| - | Without any option, the commands display all BGP peers in all instances in a summary table format. |
| summary | Displays scaled BGP peer status. |
| detail | Detailed information on all BGP peers in all instances in a list view. |
| <peer-name> | Detailed information on the peer with the given name. |
| history | Displays BGP peer history information such as the peer state down reasons. |
| history <peer-address> | Displays BGP peer history information such as the peer state down reasons for a specified peer. |
| history <peer-address> detail` | Displays Detailed BGP peer history information such as the peer state down reasons for a specified peer. |
| history detail | Displays detailed BGP history information such as the peer state down reasons. |
| address <peer-address> | Detailed information on the peer with the given IP address. |
| instance <instance-name> | Summary of all BGP peers in the given instance. |
| instance <instance-name> detail | Detailed information on all BGP peers in the given instance. |

| Option | Description |
|---|---|
| `instance <instance-name> detail <peer-name>` | Detailed information on the peer with the given name in the given instance. |
| `instance <instance-name> detail address <peer-address>` | Detailed information on the peer with the given IP address in the given instance. |
| `statistics` | Received and sent BGP prefixes per AFI/SAFI for all peers in all instances. |
| `statistics peer <peer-name>` | Received and sent BGP prefixes per AFI/SAFI for the peer with the given name. |
| `statistics peer address <peer-address>` | Received and sent BGP prefixes per AFI/SAFI for the peer with the given IP address. |
| `statistics instance <instance-name> peer <peer-name>` | Received and sent BGP prefixes per AFI/SAFI for the peer with the given name in the given instance. |
| `statistics instance <instance-name> peer address <peer-address>` | Received and sent BGP prefixes per AFI/SAFI for the peer with the given IP address in the given instance. |

Although 6PE routes are labeled, they are handled as unicast routes, and therefore will be shown as IPv6 unicast in the BGP peer statistics.

Example 1: BGP Peer Summary View

```
supervisor@rtbrick: op> show bgp peer
Instance: default
  Peer                                    Remote AS    State         Up/Down Time
PfxRcvd              PfxSent
  PE2                                     65542        Established   0d:00h:00m:13s
16                   26
```

Example 2: BGP peer summary

```
supervisor@rtbrick: op> show bgp peer summary
Instance: default
 Established: 1, OpenConfirm: 0, OpenSent: 0, Active: 0, Connect: 0, Idle: 0
```

Example 3: BGP Peer Detail View

```
supervisor@rtbrick: op> show bgp peer detail
Peer: PE2, Peer IP: 192.0.2.2, Remote AS: 65542, Local IP: 192.0.2.1, Local AS: 65541,
Any AS: False
  Type: ebgp, State: Established, Up/Down Time: Wed Jul 15 18:59:20 GMT +0000 2026
  Discovered on interface: -
  Last transition: Wed Jul 15 18:59:20 GMT +0000 2026, Flap count: 0
  Peer ID        : 198.51.100.21, Local ID: 198.51.100.11
```

```
  Instance       : default, Peer group: PE2
  6PE enabled    : False
  TTL security   : False, TTL limit: -
  Authentication:
    ID: BGP_AUTH_HMAC-SHA-256-128, Algorithm: HMAC-SHA-256-128
  Timer values:
    Peer keepalive : 30s, Local keepalive: 30s
    Peer holddown  : 90s, Local holddown : 90s
    Connect retry  : 30s
  Timers:
    Connect retry timer : 0
    Keepalive timer     : expires in 7s 263206us
    Holddown timer      : expires in 1m 14s 766679us
  NLRIs:
    Sent           : ['ipv4-unicast', 'ipv6-unicast', 'ipv4-labeled-unicast', 'ipv6-
labeled-unicast']
    Received       : ['ipv4-unicast', 'ipv6-unicast', 'ipv4-labeled-unicast', 'ipv6-
labeled-unicast']
    Negotiated     : ['ipv4-unicast', 'ipv6-unicast', 'ipv4-labeled-unicast', 'ipv6-
labeled-unicast']
  Capabilities:
    Addpath sent                  : None
    Addpath received              : None
    Addpath negotiated            : None
    Extended nexthop sent         : None
    Extended nexthop received     : None
    Extended nexthop negotiated   : None
    Multiple labels sent          : None
    Multiple labels received      : None
    Multiple labels negotiated    : None
    Capabilities:
      Feature                      Sent            Received        Negotiated
      Route refresh                True            True            True
      4 byte AS                    True            True            True
      Graceful restart             False           False           False
      Link local only              False           False           False
  Prefix Limit:
  End of RIB:
    Address family                 Sent                                Received
    IPv4 unicast                   Wed Jul 15 18:59:26 GMT +0000 2026  Wed Jul 15
18:59:26 GMT +0000 2026
    IPv4 labeled-unicast           Wed Jul 15 18:59:26 GMT +0000 2026  Wed Jul 15
18:59:26 GMT +0000 2026
    IPv6 unicast                   Wed Jul 15 18:59:26 GMT +0000 2026  Wed Jul 15
18:59:26 GMT +0000 2026
    IPv6 labeled-unicast           Wed Jul 15 18:59:26 GMT +0000 2026  Wed Jul 15
18:59:26 GMT +0000 2026
    IPv4 VPN-unicast               never                               never
    IPv6 VPN-unicast               never                               never
    IPv4 flowspec                  never                               never
    IPv6 flowspec                  never                               never
```

```
    IPv4 VPN-multicast             never                                never
    L2VPN VPLS                     never                                never
    L2VPN EVPN                     never                                never
  RPKI Validation Stats:
    Address family                 Valid    Invalid    Unknown
    IPv4 unicast                       0          0          0
    IPv6 unicast                       0          0          0
  Message stats:
    Session stats:
      Direction              Open          Update       Keepalive          Notify
Route refresh
      Input                     1               8               1               0
0
      Output                    1               8               1               0
0
    Total stats:
      Input                     1               8               1               0
0
      Output                    1               8               1               0
0
    Route stats:
      Address family                        Received            Sent    Prefix limit
Idle timeout
      IPv4 unicast                                 4               8               0
0
      IPv4 labeled-unicast                         4               6               0
0
      IPv6 unicast                                 4               6               0
0
      IPv6 labeled-unicast                         4               6               0
0
      IPv4 VPN-unicast                             0               0               0
0
      IPv6 VPN-unicast                             0               0               0
0
      IPv4 multicast                               0               0               0
0
      IPv4 VPN-multicast                           0               0               0
0
      L2VPN VPLS                                   0               0               0
0
      L2VPN EVPN                                   0               0               0
0
      IPv4 Flowspec                                0               0               0
0
      IPv6 Flowspec                                0               0               0
0
```

Example 4: BGP Peer Statistics

```
supervisor@rtbrick: op> show bgp peer statistics instance default peer PE2
Instance: default
  Peer                            AFI     SAFI                PfxRcvd    PfxSent
  PE2                             ipv4    unicast             4          8
                                  ipv4    labeled-unicast     4          6
                                  ipv6    unicast             4          6
                                  ipv6    labeled-unicast     4          6
                                  ipv4    vpn-unicast         0          0
                                  ipv6    vpn-unicast         0          0
                                  ipv4    multicast           0          0
                                  ipv4    vpn-multicast       0          0
                                  l2vpn   vpls                0          0
                                  l2vpn   evpn                0          0
                                  ipv4    flowspec            0          0
                                  ipv6    flowspec            0          0
```

Example 5: BGP Peer history for a specified peer

```
supervisor@rtbrick: op> show bgp peer history peer address 192.0.2.2
Instance: default
  Peer Address                              Source Address
Type         Last Reset Reason  Update Time
  192.0.2.2                                 192.0.2.1
Notifica...  Cease, Sub-Cod...  Wed Jul 15 18:59:40
  192.0.2.2                                 192.0.2.1
Clear BG...  Cease, Sub-Cod...  Wed Jul 15 18:59:37
```

Example 6: Detailed BGP peer history information

```
supervisor@rtbrick: op> show bgp peer history detail
Instance: default
  Peer address: 192.0.2.2, Source address: 192.0.2.1
    Type: Notification Message Received, Update time: Wed Jul 15 18:59:40 GMT +0000
2026,
    Last reset reason: Cease, Sub-Code: Connect reject
  Peer address: 192.0.2.2, Source address: 192.0.2.1
    Type: Clear BGP Neighbor, Update time: Wed Jul 15 18:59:37 GMT +0000 2026,
    Last reset reason: Cease, Sub-Code: Admin reset
```

Example 7: Detailed BGP peer information for actual sent and received route counters exchanged between BGP peers for each address family.

```
supervisor@rtbrick.net: op> show bgp peer detail
Peer: P2_SDN, Peer IP: 192.0.2.3, Remote AS: 4200000003, Local IP: 192.0.2.1, Local
AS: 4200000001, Any AS: True
  Type: ebgp, State: Established, Up/Down Time: Tue May 19 15:33:03 GMT +0000 2026
  Discovered on interface: -
```

```
  Last transition: Tue May 19 15:33:03 GMT +0000 2026, Flap count: 0
  Peer ID        : 10.0.0.30, Local ID: 10.0.0.10
  Instance       : default, Peer group: P2_SDN
  6PE enabled    : False
  TTL security   : False, TTL limit: -
  Authentication:
    ID: BGP_AUTH_HMAC-SHA-256-128, Algorithm: HMAC-SHA-256-128
  Timer values:
    Peer keepalive : 30s, Local keepalive: 30s
    Peer holddown  : 90s, Local holddown : 90s
    Connect retry  : 30s
  Timers:
    Connect retry timer : 0
    Keepalive timer     : expires in 22s 390625us
    Holddown timer      : expires in 1m 9s 684574us
  NLRIs:
    Sent           : ['ipv4-unicast', 'ipv6-unicast', 'ipv4-labeled-unicast', 'ipv6-
labeled-unicast']
    Received       : ['ipv4-unicast', 'ipv6-unicast', 'ipv4-labeled-unicast', 'ipv6-
labeled-unicast']
    Negotiated     : ['ipv4-unicast', 'ipv6-unicast', 'ipv4-labeled-unicast', 'ipv6-
labeled-unicast']
  Capabilities:
    Addpath sent                  : None
    Addpath received              : None
    Addpath negotiated            : None
    Extended nexthop sent         : None
    Extended nexthop received     : None
    Extended nexthop negotiated   : None
    Multiple labels sent          : ['ipv4-labeled-unicast', 'ipv6-labeled-unicast']
    Multiple labels received      : ['ipv4-labeled-unicast', 'ipv6-labeled-unicast']
    Multiple labels negotiated    : ['ipv4-labeled-unicast', 'ipv6-labeled-unicast']
    Capabilities:
      Feature                       Sent            Received        Negotiated
      Route refresh                True            True            True
      4 byte AS                    True            True            True
      Graceful restart             False           False           False
      Link local only              False           False           False
  Prefix Limit:
  End of RIB:
    Address family                 Sent                                Received
    IPv4 unicast                    Tue May 19 15:33:09 GMT +0000 2026  Tue May 19
15:33:08 GMT +0000 2026
    IPv4 labeled-unicast            Tue May 19 15:33:09 GMT +0000 2026  Tue May 19
15:33:08 GMT +0000 2026
    IPv6 unicast                    Tue May 19 15:33:09 GMT +0000 2026  Tue May 19
15:33:08 GMT +0000 2026
    IPv6 labeled-unicast            Tue May 19 15:33:09 GMT +0000 2026  Tue May 19
15:33:08 GMT +0000 2026
    IPv4 VPN-unicast                never                               never
    IPv6 VPN-unicast                never                               never
```

```
    IPv4 flowspec                   never                                never
    IPv6 flowspec                   never                                never
    IPv4 VPN-multicast              never                                never
    L2VPN VPLS                      never                                never
    L2VPN EVPN                      never                                never
  RPKI Validation Stats:
    Address family                 Valid    Invalid    Unknown
    IPv4 unicast                       0          0          0
    IPv6 unicast                       0          0          0
  Message stats:
    Session stats:
      Direction              Open         Update       Keepalive         Notify  Route
refresh
      Input                     1             12            1918                0
0
      Output                    1              8            1918                0
0
    Total stats:
      Input                     1             12            1918                0
0
      Output                    1              8            1918                0
0
    Route stats:
      Address family                     Received           Sent    Prefix limit
Idle timeout
      IPv4 unicast                              4              7               0
0
      IPv4 labeled-unicast                      6              9               0
0
      IPv6 unicast                              4              7               0
0
      IPv6 labeled-unicast                      6              9               0
0
      IPv4 VPN-unicast                          0              0               0
0
      IPv6 VPN-unicast                          0              0               0
0
      IPv4 multicast                            0              0               0
0
      IPv4 VPN-multicast                        0              0               0
0
      L2VPN VPLS                                0              0               0
0
      L2VPN EVPN                                0              0               0
0
      IPv4 Flowspec                             0              0               0
0
```

To access the Operational State API that corresponds to this CLI, click here.

#### BGP Peer Group

The 'show bgp peer-group' commands display parameters like BGP attributes that are specific to the respective peer groups.

**Syntax:**

**show bgp peer-group** <option> ...

| Option | Description |
|---|---|
| - | Without any option, the commands display information on all peer groups in all instances. |
| <peer-group-name> | Information on the peer group with the given name. |
| instance <instance-name> | All peer groups in the given instance. |
| instance <instance-name> <peer-group-name> | Information on the peer group with the given name in the given instance. |

Example 1: BGP Peer Group

```
supervisor@rtbrick: op> show bgp peer-group PE2
Instance: default
  Peer group name: PE2
    Remote AS: 65542
```

Example 2: BGP peer group with TTL limit

```
supervisor@rtbrick: op> show bgp peer-group
Instance: default
  Peer group name: PE2
    Remote AS: 65542
    TTL security: True, TTL limit: 4
```

Example 3: BGP peer group with EBGP multihop

```
supervisor@rtbrick: op> show bgp peer-group
Instance: default
  Peer group name: PE2
    Remote AS: 65542, EBGP multihop: 2
```

#### BGP FIB

The `show bgp fib` commands display the BGP forwarding table. In contrast to the 'show bgp rib' commands, the output of the `show bgp fib` commands includes only the selected routes. The BGP route selection occurs between the RIB and the FIB.

**Syntax:**

**show bgp fib** <option> ...

| Option | Description |
|---|---|
| - | Without any option, the commands display the BGP forwarding table for all address families and all instances in a summary table format. |
| instance | Instance name. |
| summary | Displays summary information. |
| <afi> | BGP forwarding table summary for the given address family (AFI), all sub-address families and all instances. Supported AFI values are 'ipv4' and 'ipv6'. |
| <afi> <safi> | BGP forwarding table summary for the given address family (AFI) and sub-address family (SAFI), and all instances. Supported SAFI values are 'unicast', 'labeled-unicast', 'multicast', 'vpn-multicast', and 'vpn-unicast'. |
| <afi> <safi> detail | Detailed list view of the BGP forwarding table for the given address family (AFI) and sub-address family (SAFI), and all instances. |
| <afi> <safi> <prefix> | BGP forwarding table entry for the given prefix and all instances. |
| <afi> <safi> instance <instance-name> | BGP forwarding table summary for the given AFI, SAFI, and instance. |
| <afi> <safi> instance <instance-name> detail | Detailed list view of BGP forwarding table for the given AFI, SAFI, and instance. |
| <afi> <safi> instance <instance-name> <prefix> | BGP forwarding table entry for the given prefix and instance. |

Example 1: Summary information for the installed BGP routes in the FIB under a specific BGP instance.

```
supervisor@rtbrick: op> show bgp fib instance default summary
Instance: default, Total Routes: 8
  Address family                 Routes
  IPv4 unicast                        2
  IPv4 labeled-unicast                2
  IPv6 unicast                        2
  IPv6 labeled-unicast                2
  Total Routes                        8
```

Example 2: Summary view of the BGP FIB for IPv6, all SAFIs and all instances

```
supervisor@rtbrick: op> show bgp fib ipv6
Instance: default, AFI: ipv6, SAFI: unicast, Total routes: 2
  Prefix                                        Preference      Label
Next Hop
  2001:db8:0:1::21/128                          20              -
192.0.2.2
  2001:db8:0:1::22/128                          20              -
192.0.2.2
Instance: default, AFI: ipv6, SAFI: labeled-unicast, Total routes: 2
  Prefix                                        Preference      Label
Next Hop
  2001:db8:0:1::21/128                          20              -
192.0.2.2
  2001:db8:0:1::22/128                          20              -
192.0.2.2
```

Example 3: Detailed view of the BGP FIB for IPv6 VPN unicast routes in the default instances

```
supervisor@rtbrick: op> show bgp fib ipv6 unicast instance default detail
Instance: default, AFI: ipv6, SAFI: unicast, Total routes: 2
  Prefix: 2001:db8:0:1::21/128
    Next hop key: 4817d0c3f4cdff043db9a1b609d636032d017cef5a6733de
    Route source: bgp, Send path ID: 185972253, Received path ID: None
    AS path: [65542], Originator ID: None, Origin: Incomplete
    Community: None
    Extended community: None
    Large community: None
    Cluster list: None
    IGP metric: 4294967295, Local preference: None, Multi exit discriminator: 0
    Preference: 20, External route: None, Readvertised route: None
    Route up: None
    Next hop: 192.0.2.2, Label: -
  Prefix: 2001:db8:0:1::22/128
    Next hop key: 4817d0c3f4cdff043db9a1b609d636032d017cef5a6733de
    Route source: bgp, Send path ID: 185972253, Received path ID: None
    AS path: [65542], Originator ID: None, Origin: Incomplete
    Community: None
    Extended community: None
    Large community: None
    Cluster list: None
    IGP metric: 4294967295, Local preference: None, Multi exit discriminator: 0
    Preference: 20, External route: None, Readvertised route: None
    Route up: None
    Next hop: 192.0.2.2, Label: -
```

Example 4: Summary information for the installed BGP routes in the FIB under a specific BGP instance.

```
supervisor@rtbrick: op> show bgp fib instance default summary
Instance: default, Total Routes: 8
  Address family                 Routes
  IPv4 unicast                        2
  IPv4 labeled-unicast                2
  IPv6 unicast                        2
  IPv6 labeled-unicast                2
  Total Routes                        8
```

Example 5: Summary information for the installed BGP routes in the FIB for a specified AFI and SAFI within a specific BGP instance.

```
supervisor@rtbrick: op> show bgp fib ipv6 unicast summary
Instance: default, Total Routes: 2
  Address family                 Routes
  IPv6 unicast                        2
  Total Routes                        2
```

Example 5: Summary information for the installed BGP routes in the FIB.

```
supervisor@rtbrick: op> show bgp fib summary
Instance: default, Total Routes: 8
  Address family                 Routes
  IPv4 unicast                        2
  IPv4 labeled-unicast                2
  IPv6 unicast                        2
  IPv6 labeled-unicast                2
  Total Routes                        8
```

#### BGP RIB-in

This command displays information about the BGP routes received from BGP neighbors.

**Syntax:**

**show bgp rib-in** <option>

| Attribute | Description |
|---|---|
| `-` | Without any option, the command displays information on the received BGP routing table on all instances in a summary table format. |
| `summary` | Displays a summary of the RIB for all instances. |

| Attribute | Description |
| --- | --- |
| `instance <instance-name>` | Name of the instance. Displays the RIB summary information for BGP routes received from all neighbors for a specified instance. Additional filtering options such as `detail`, `prefix`, `peer`, `peer address`, and `summary` can be added to further narrow down the results. |
| `instance <instance_name> peer address <peer_ip>` | Displays the RIB entry for BGP routes learned from a specific peer IP address for a specified instance. Additional filtering options such as `detail`, `summary` can be used to further narrow down the results. |
| `instance <instance_name> peer <peer-name>` | Displays the RIB entry for BGP routes learned from a specific peer (name) for a specified instance. Additional filtering options such as `detail`, `summary` can be used to further narrow down the results. |
| `instance <instance_name> <prefix>` | Displays the RIB entry for a given prefix for a specified instance. Additional filtering options such as `detail` can be used to further narrow down the results. |
| `instance <instance_name> summary` | Displays a summary of the RIB for a specified instance. |
| `<prefix>` | Displays the RIB entry for a given prefix across all instances. Additional filtering options such as `detail` can be used to further narrow down the results. |
| `<afi>` | BGP routing table summary for the given address family (AFI), all sub-address families and all instances. Supported AFI values are 'ipv4' and 'ipv6'. Additional filtering options such as `detail`, `flowspec`, and `instance` can be added to further narrow down the results. |
| `<afi> detail` | Detailed list view of the BGP routing table for the given AFI, |
| `<afi> flowspec` | Displays BGP routing table entries for FlowSpec routes for the given AFI, all sub-address families and all instances. Supported AFI values are 'ipv4' and 'ipv6'. Additional filtering options such as `community`, `detail`, `filter`, `hash`, `instance`, `peer address`, `peer`, and `summary` can be added to further narrow down the results. |
| `<afi> <prefix>` | Displays the RIB entry for a given prefix for a specified address family. Supported AFI values are 'ipv4' and 'ipv6'. Additional filtering options such as `detail` can be added to further narrow down the results. |
| `<afi> <safi>` | Displays BGP routing table summary for the given AFI and subsequent address family identifier (SAFI), and all instances. Supported SAFI values are 'labeled-unicast', 'unicast', 'vpn-multicast', and 'vpn-unicast'. Additional filtering options such as `community`, `detail`, `instance`, `peer address`, `peer`, `source`, and `summary` can be added to further narrow down the results. |

| Attribute | Description |
| --- | --- |
| `<afi> <safi> community <community>` | Displays BGP routing table entries for a given community for the specified AFI and SAFI, and all instances. Additional filtering options such as `detail` can be added to further narrow down the results. |
| `<afi> <safi> detail` | Detailed list view of the BGP routing table for the given AFI and SAFI, and all instances. Supported SAFI values are 'labeled-unicast', 'unicast', 'vpn-multicast', and 'vpn-unicast'. |
| `<afi> <safi> instance <instance_name>` | BGP routing table summary for the given AFI, SAFI, and instance. Supported SAFI values are 'labeled-unicast', 'unicast', 'vpn-multicast', and 'vpn-unicast'. Additional filtering options such as `detail`, `peer address`, `peer`, `prefix`, and `summary` can be added to further narrow down the results. |
| `<afi> <safi>source <source>` | Displays a summarized list of all BGP routes received from a peer (source) for a specified AFI and SAFI. Supported AFI values are 'ipv4' and 'ipv6'. Additional filtering options such as `detail` can be added to further narrow down the results. |
| `post-policy` | Displays the post-policy RIB summary information for BGP routes received from all neighbors for all instances. Additional filtering options such as `instance`, `import`, `prefix`, `source`, `summary`, and `validation-state` can be added to further narrow down the results. |
| `post-policy summary` | Displays a summary of the post-policy RIB. |
| `post-policy instance <instance-name>` | Name of the instance. Displays the post-policy RIB summary information for BGP routes received from all neighbors for a specified instance. Additional filtering options such as `detail`, `error`, `import`, `peer`, `peer address`, `policy-accept`, `policy-deny`, `summary`, and `validation-state` can be added to further narrow down the results. |
| `post-policy instance <instance-name> error` | Displays the post-policy RIB entry for BGP routes that encountered errors during processing for a specified instance. Additional filtering options such as `detail`, `nexthop-unreachable`, and `retain-rt-deny` can be used to further narrow down the results. |
| `post-policy instance <instance-name> import` | Displays the post-policy RIB entry for BGP routes that were rejected by the import policy for a specified instance. Additional filtering options such as `detail` and `summary` can be used to further narrow down the results. |
| `post-policy instance <instance-name> peer address <peer_ip>` | Displays the post-policy RIB entry for BGP routes learned from a specific peer IP address for a specified instance. Additional filtering options such as `detail`, `error`, `policy-accept`, `policy-deny`, `summary` can be used to further narrow down the results. |

| Attribute | Description |
| --- | --- |
| `post-policy instance <instance-name> peer <peer-name>` | Displays the post-policy RIB entry for BGP routes learned from a specific peer (name) for a specified instance. Additional filtering options such as `detail`, `error`, `policy-accept`, `policy-deny`, `summary` can be used to further narrow down the results. |
| `post-policy instance <instance-name> policy-accept` | Displays the post-policy RIB entry for BGP routes that were accepted by the import policy for a specified instance. Additional filtering options such as `detail` can be used to further narrow down the results. |
| `post-policy instance <instance-name> policy-deny` | Displays the post-policy RIB entry for BGP routes that were denied by the import policy for a specified instance. Additional filtering options such as `detail` can be used to further narrow down the results. |
| `post-policy instance <instance-name> <prefix>` | Displays the post-policy RIB entry for a given prefix for a specified instance. Additional filtering options such as `detail` can be used to further narrow down the results. |
| `post-policy instance <instance-name> summary` | Displays a summary of the post-policy RIB for a specified instance. |
| `post-policy instance <instance-name> validation-state <state>` | Displays the post-policy RIB entry for BGP routes with a specific validation state for a specified instance. Supported validation states are `unknown`, `valid`, `invalid`, and `init`. Additional filtering options such as `detail` can be added to further narrow down the results. |
| `post-policy <prefix>` | Displays the post-policy RIB entry for a given prefix for all instances. Additional filtering options such as `detail` can be added to further narrow down the results. |
| `post-policy <afi>` | BGP routing table summary for the given AFI, all sub-address families and all instances. Supported AFI values are 'ipv4' and 'ipv6'. Additional filtering options such as `detail`, `flowspec`, and `instance` can be added to further narrow down the results. |
| `post-policy <afi> detail` | Detailed list view of the BGP routing table for the given AFI. |
| `post-policy <afi> flowspec` | Displays BGP routing table entries for FlowSpec routes for the given AFI, all sub-address families and all instances. Supported AFI values are 'ipv4' and 'ipv6'. Additional filtering options such as `community`, `detail`, `error`, `filter`, `hash`, `import`, `instance`, `peer address`, `peer`, `policy-accept`, `policy-deny`, and `summary` can be added to further narrow down the results. |
| `post-policy <afi> <prefix>` | Displays the post-policy RIB entry for a given prefix for a specified AFI. Supported AFI values are 'ipv4' and 'ipv6'. Additional filtering options such as `detail` can be added to further narrow down the results. |

| Attribute | Description |
|---|---|
| `post-policy <afi> source <source>` | Displays a summarized list of all BGP routes received from a peer (source) for a specified AFI. Supported AFI values are 'ipv4' and 'ipv6'. Additional filtering options such as `detail` can be added to further narrow down the results. |
| `post-policy <afi> <safi>` | Displays BGP routing table summary for the given AFI and SAFI, and all instances. Supported SAFI values are 'labeled-unicast', 'unicast', 'vpn-multicast', and 'vpn-unicast'. Additional filtering options such as `community`, `detail`, `error`, `import`, `instance`, `peer address`, `peer`, `policy-accept`, `policy-deny` can be added to further narrow down the results. |
| `post-policy <afi> <safi> community <community>` | Displays BGP routing table entries for a given community for the specified AFI and SAFI, and all instances. Additional filtering options such as `detail` can be added to further narrow down the results. |
| `post-policy <afi> <safi> error` | BGP routes with error status for the given AFI and SAFI, and all instances. Supported SAFI values are 'labeled-unicast', 'unicast', 'vpn-multicast', and 'vpn-unicast'. Additional filtering options such as `detail`, `nexthop-unreachable`, and `retain-rt-deny` can be used to further narrow down the results. |
| `post-policy <afi> <safi> import` | BGP routes that were rejected by the import policy for the given AFI and SAFI, and all instances. Supported SAFI values are 'labeled-unicast', 'unicast', 'vpn-multicast', and 'vpn-unicast'. Additional filtering options such as `detail`, `summary` and `validation-state` can be used to further narrow down the results. |
| `post-policy <afi> <safi> detail` | Detailed list view of the BGP routing table for the given AFI and SAFI, and all instances. Supported SAFI values are 'labeled-unicast', 'unicast', 'vpn-multicast', and 'vpn-unicast'. |
| `post-policy <afi> <safi> instance <instance_name>` | BGP routing table summary for the given AFI, SAFI, and instance. Supported SAFI values are 'labeled-unicast', 'unicast', 'vpn-multicast', and 'vpn-unicast'. Additional filtering options such as `detail`, `error`, `import`, `peer address`, `peer`, `policy-accept`, `policy-deny`, and `summary` can be added to further narrow down the results. |

Example 1: BGP received routes

```
supervisor@rtbrick: op> show bgp rib-in
Instance: default, AFI: ipv4, SAFI: unicast
  Hostname: Local, Peer IP: 0.0.0.0
  Source IP: 0.0.0.0, Total routes: 6
    Prefix                                      Next Hop
MED         Lpref       AS Path
    192.0.2.0/30                                -                                   0
-           -
```

```
    192.0.2.4/30                                  -                                 0
-           -
    198.51.100.11/32                              -                                 0
-           -
    198.51.100.12/32                              -                                 0
-           -
    198.51.100.100/32                             -                                 0
-           -
    198.51.100.101/32                             -                                 0
-           -
Instance: default, AFI: ipv4, SAFI: unicast
  Hostname: PE2, Peer IP: 192.0.2.2
  Source IP: 192.0.2.1, Total routes: 4
    Prefix                                        Next Hop
MED         Lpref       AS Path
    192.0.2.0/30                                  192.0.2.2                         0
-           65542
    192.0.2.4/30                                  192.0.2.2                         0
-           65542
    198.51.100.21/32                              192.0.2.2                         0
-           65542
    198.51.100.22/32                              192.0.2.2                         0
-           65542
Instance: default, AFI: ipv4, SAFI: labeled-unicast
  Hostname: Local, Peer IP: 0.0.0.0
  Source IP: 0.0.0.0, Total routes: 4
    Prefix                                        Next Hop
MED         Lpref       AS Path
    192.0.2.0/30                                  -                                 0
-           -
    192.0.2.4/30                                  -                                 0
-           -
    198.51.100.11/32                              -                                 0
-           -
    198.51.100.12/32                              -                                 0
-           -
Instance: default, AFI: ipv4, SAFI: labeled-unicast
  Hostname: PE2, Peer IP: 192.0.2.2
  Source IP: 192.0.2.1, Total routes: 4
    Prefix                                        Next Hop
MED         Lpref       AS Path
    192.0.2.0/30                                  192.0.2.2                         0
-           65542
    192.0.2.4/30                                  192.0.2.2                         0
-           65542
    198.51.100.21/32                              192.0.2.2                         0
-           65542
    198.51.100.22/32                              192.0.2.2                         0
-           65542
Instance: default, AFI: ipv6, SAFI: unicast
  Hostname: PE2, Peer IP: 192.0.2.2
```

```
  Source IP: 192.0.2.1, Total routes: 4
    Prefix                                      Next Hop
MED         Lpref       AS Path
    2001:db8:1:1::/64                           192.0.2.2                          0
-           65542
    2001:db8:2:1::/64                           192.0.2.2                          0
-           65542
    2001:db8:0:1::21/128                        192.0.2.2                          0
-           65542
    2001:db8:0:1::22/128                        192.0.2.2                          0
-           65542
Instance: default, AFI: ipv6, SAFI: unicast
  Hostname: Local, Peer IP: ::
  Source IP: ::, Total routes: 4
    Prefix                                      Next Hop
MED         Lpref       AS Path
    2001:db8:1:1::/64                           -                                  0
-           -
    2001:db8:2:1::/64                           -                                  0
-           -
    2001:db8:0:1::11/128                        -                                  0
-           -
    2001:db8:0:1::12/128                        -                                  0
-           -
Instance: default, AFI: ipv6, SAFI: labeled-unicast
  Hostname: PE2, Peer IP: 192.0.2.2
  Source IP: 192.0.2.1, Total routes: 4
    Prefix                                      Next Hop
MED         Lpref       AS Path
    2001:db8:1:1::/64                           192.0.2.2                          0
-           65542
    2001:db8:2:1::/64                           192.0.2.2                          0
-           65542
    2001:db8:0:1::21/128                        192.0.2.2                          0
-           65542
    2001:db8:0:1::22/128                        192.0.2.2                          0
-           65542
Instance: default, AFI: ipv6, SAFI: labeled-unicast
  Hostname: Local, Peer IP: ::
  Source IP: ::, Total routes: 4
    Prefix                                      Next Hop
MED         Lpref       AS Path
    2001:db8:1:1::/64                           -                                  0
-           -
    2001:db8:2:1::/64                           -                                  0
-           -
    2001:db8:0:1::11/128                        -                                  0
-           -
    2001:db8:0:1::12/128                        -                                  0
-           -
```

Example 2: Summary of the BGP received routes

```
supervisor@rtbrick: op> show bgp rib-in summary
Instance: default, Total Routes: 34
  Hostname: Local, Peer IP: 0.0.0.0, Source IP: 0.0.0.0
    Address family                 Routes
    IPv4 unicast                        6
    IPv4 labeled-unicast                4
    Total Routes                       10
  Hostname: PE2, Peer IP: 192.0.2.2, Source IP: 192.0.2.1
    Address family                 Routes
    IPv4 unicast                        4
    IPv4 labeled-unicast                4
    IPv6 unicast                        4
    IPv6 labeled-unicast                4
    Total Routes                       16
  Hostname: Local, Peer IP: ::, Source IP: ::
    Address family                 Routes
    IPv6 unicast                        4
    IPv6 labeled-unicast                4
    Total Routes                        8
```

Example 3: View of BGP routes filtered based in the source protocol, direct.

```
supervisor@rtbrick: op> show bgp rib-in source direct
Instance: default, AFI: ipv4, SAFI: unicast
  Hostname: Local, Peer IP: 0.0.0.0
  Source IP: 0.0.0.0, Total routes: 4
    Prefix                                      Next Hop
MED         Lpref       AS Path
    192.0.2.0/30                                -                                   0
-           -
    192.0.2.4/30                                -                                   0
-           -
    198.51.100.11/32                            -                                   0
-           -
    198.51.100.12/32                            -                                   0
-           -
Instance: default, AFI: ipv4, SAFI: labeled-unicast
  Hostname: Local, Peer IP: 0.0.0.0
  Source IP: 0.0.0.0, Total routes: 4
    Prefix                                      Next Hop
MED         Lpref       AS Path
    192.0.2.0/30                                -                                   0
-           -
    192.0.2.4/30                                -                                   0
-           -
    198.51.100.11/32                            -                                   0
-           -
```

```
    198.51.100.12/32                            -                                0
-           -
Instance: default, AFI: ipv6, SAFI: unicast
  Hostname: Local, Peer IP: ::
  Source IP: ::, Total routes: 4
    Prefix                                      Next Hop
MED         Lpref       AS Path
    2001:db8:1:1::/64                           -                                0
-           -
    2001:db8:2:1::/64                           -                                0
-           -
    2001:db8:0:1::11/128                        -                                0
-           -
    2001:db8:0:1::12/128                        -                                0
-           -
Instance: default, AFI: ipv6, SAFI: labeled-unicast
  Hostname: Local, Peer IP: ::
  Source IP: ::, Total routes: 4
    Prefix                                      Next Hop
MED         Lpref       AS Path
    2001:db8:1:1::/64                           -                                0
-           -
    2001:db8:2:1::/64                           -                                0
-           -
    2001:db8:0:1::11/128                        -                                0
-           -
    2001:db8:0:1::12/128                        -                                0
-           -
```

Example 4: Summary view of the BGP rib-in for IPv4, all SAFIs and all instances

```
supervisor@rtbrick: op> show bgp rib-in ipv4
Instance: default, AFI: ipv4, SAFI: unicast
  Hostname: Local, Peer IP: 0.0.0.0
  Source IP: 0.0.0.0, Total routes: 6
    Prefix                                      Next Hop
MED         Lpref       AS Path
    192.0.2.0/30                                -                                0
-           -
    192.0.2.4/30                                -                                0
-           -
    198.51.100.11/32                            -                                0
-           -
    198.51.100.12/32                            -                                0
-           -
    198.51.100.100/32                           -                                0
-           -
    198.51.100.101/32                           -                                0
-           -
Instance: default, AFI: ipv4, SAFI: unicast
```

```
  Hostname: PE2, Peer IP: 192.0.2.2
  Source IP: 192.0.2.1, Total routes: 4
    Prefix                                      Next Hop
MED         Lpref       AS Path
    192.0.2.0/30                                192.0.2.2                          0
-           65542
    192.0.2.4/30                                192.0.2.2                          0
-           65542
    198.51.100.21/32                            192.0.2.2                          0
-           65542
    198.51.100.22/32                            192.0.2.2                          0
-           65542
Instance: default, AFI: ipv4, SAFI: labeled-unicast
  Hostname: Local, Peer IP: 0.0.0.0
  Source IP: 0.0.0.0, Total routes: 4
    Prefix                                      Next Hop
MED         Lpref       AS Path
    192.0.2.0/30                                -                                  0
-           -
    192.0.2.4/30                                -                                  0
-           -
    198.51.100.11/32                            -                                  0
-           -
    198.51.100.12/32                            -                                  0
-           -
Instance: default, AFI: ipv4, SAFI: labeled-unicast
  Hostname: PE2, Peer IP: 192.0.2.2
  Source IP: 192.0.2.1, Total routes: 4
    Prefix                                      Next Hop
MED         Lpref       AS Path
    192.0.2.0/30                                192.0.2.2                          0
-           65542
    192.0.2.4/30                                192.0.2.2                          0
-           65542
    198.51.100.21/32                            192.0.2.2                          0
-           65542
    198.51.100.22/32                            192.0.2.2                          0
-           65542
```

Example 5: Summary view of the received routes

```
supervisor@rtbrick: op> show bgp rib-in ipv4 unicast instance default peer address
192.0.2.2
Instance: default, AFI: ipv4, SAFI: unicast
  Hostname: PE2, Peer IP: 192.0.2.2
  Source IP: 192.0.2.1, Total routes: 4
    Prefix                                      Next Hop
MED         Lpref       AS Path
    192.0.2.0/30                                192.0.2.2                          0
-           65542
```

```
    192.0.2.4/30                                192.0.2.2                        0
-           65542
    198.51.100.21/32                            192.0.2.2                        0
-           65542
    198.51.100.22/32                            192.0.2.2                        0
-           65542
```

Example 6: Summary view of the BGP rib-in post-policy

```
supervisor@rtbrick.net: op> show bgp rib-in post-policy
Flags: & - Imported, ! - Error, N - RPKI Unknown, I - RPKI Invalid, V - RPKI Valid
Instance: default, AFI: ipv4, SAFI: unicast
  Hostname: Local, Peer IP: 0.0.0.0
  Source IP: 0.0.0.0, Total routes: 9
    Flags  Prefix                                Next Hop          MED         Lpref
AS Path
           192.0.2.0/24                          -                 0           -
-
           192.0.2.1/24                          -                 0           -
-
           192.0.2.2/24                          -                 0           -
-
           203.0.113.20/32                       -                 0           -
-
           203.0.113.21/32                       -                 0           -
-
           203.0.113.22/32                       -                 0           -
-
           203.0.113.23/32                       -                 0           -
-
           203.0.113.24/32                       -                 0           -
-
           203.0.113.25/32                       -                 0           -
-
Instance: default, AFI: ipv4, SAFI: unicast
  Hostname: PE1, Peer IP: 192.0.2.1
  Source IP: 192.0.2.2, Total routes: 3
    Flags  Prefix                                Next Hop          MED         Lpref
AS Path
           192.0.2.10/32                         192.0.2.1         0           -
65536
           192.0.2.11/32                         192.0.2.1         0           -
65536
           192.0.2.12/32                         192.0.2.1         0           -
65536
Instance: default, AFI: ipv4, SAFI: unicast
  Hostname: PE1, Peer IP: 2001:DB8::1
  Source IP: 2001:DB8::2, Total routes: 3
    Flags  Prefix                                Next Hop          MED         Lpref
AS Path
```

```
           192.0.2.10/32                        2001:DB8::1       0           -
65536
           192.0.2.11/32                        2001:DB8::1       0           -
65536
           192.0.2.12/32                        2001:DB8::1       0           -
65536
Instance: vrf1-blue, AFI: ipv4, SAFI: unicast
  Hostname: Local, Peer IP: 0.0.0.0
  Source IP: 0.0.0.0, Total routes: 5
    Flags  Prefix                               Next Hop          MED         Lpref
AS Path
           0.0.0.0/0                            -                 0           -
-
           192.0.2.30/24                        -                 0           -
-
           192.0.2.31/32                        -                 0           -
-
           192.0.2.32/32                        -                 0           -
-
           192.0.2.33/32                        -                 0           -
-
```

Example 7: Summary view of the BGP rib-in post-policy for a specific instance, AFI, and SAFI.

```
supervisor@rtbrick.net: op> show bgp rib-in post-policy ipv4 unicast instance default
Flags: & - Imported, ! - Error, N - RPKI Unknown, I - RPKI Invalid, V - RPKI Valid
Instance: default, AFI: ipv4, SAFI: unicast
  Hostname: Local, Peer IP: 0.0.0.0
  Source IP: 0.0.0.0, Total routes: 9
    Flags  Prefix                               Next Hop            MED         Lpref
AS Path
           192.0.2.0/24                         -                   0           -
-
           192.0.2.1/24                         -                   0           -
-
           192.0.2.2/24                         -                   0           -
-
           192.0.2.3/32                         -                   0           -
-
           192.0.2.4/32                         -                   0           -
-
           192.0.2.5/32                         -                   0           -
-
           192.0.2.6/32                         -                   0           -
-
           192.0.2.7/32                         -                   0           -
-
           192.0.2.8/32                         -                   0           -
-
Instance: default, AFI: ipv4, SAFI: unicast
```

```
  Hostname: PE1, Peer IP: 192.0.2.9
  Source IP: 192.0.2.10, Total routes: 3
    Flags  Prefix                              Next Hop           MED         Lpref
AS Path
           192.0.2.11/32                       192.0.2.9          0           -
65536
           192.0.2.12/32                       192.0.2.9          0           -
65536
           192.0.2.13/32                       192.0.2.9          0           -
65536
Instance: default, AFI: ipv4, SAFI: unicast
  Hostname: PE1, Peer IP: 192.0.2.14
  Source IP: 192.0.2.15, Total routes: 3
    Flags  Prefix                              Next Hop           MED         Lpref
AS Path
           192.0.2.11/32                       192.0.2.14         0           -
65536
           192.0.2.12/32                       192.0.2.14         0           -
65536
           192.0.2.13/32                       192.0.2.14         0           -
65536
```

Example 8: Summary view of the BGP rib-in post-policy for a specific instance, AFI, SAFI, and policy-accept filter.

```
supervisor@rtbrick.net: op> show bgp rib-in post-policy ipv4 unicast policy-accept
Flags: & - Imported, ! - Error, N - RPKI Unknown, I - RPKI Invalid, V - RPKI Valid
Instance: default, AFI: ipv4, SAFI: unicast
  Hostname: PE1, Peer IP: 192.0.2.1
  Source IP: 192.0.2.2, Total routes: 3
    Flags  Prefix                              Next Hop           MED         Lpref
AS Path
           192.168.1.10/32                     192.0.2.1          0           -
65536
           192.168.2.10/32                     192.0.2.1          0           -
65536
           192.168.1.11/32                     192.0.2.1          0           -
65536
Instance: default, AFI: ipv4, SAFI: unicast
  Hostname: PE1, Peer IP: 192::1
  Source IP: 192::2, Total routes: 3
    Flags  Prefix                              Next Hop           MED         Lpref
AS Path
           192.168.1.10/32                     192::1             0           -
65536
           192.168.2.10/32                     192::1             0           -
65536
           192.168.1.11/32                     192::1             0           -
65536
```

Example 9: Summary view of the BGP rib-in post-policy for a specific instance, AFI, SAFI, and policy-deny filter.

```
supervisor@rtbrick.net: op> show bgp rib-in post-policy ipv4 unicast policy-deny
Flags: & - Imported, ! - Error, N - RPKI Unknown, I - RPKI Invalid, V - RPKI Valid
Instance: default, AFI: ipv4, SAFI: unicast
  Hostname: PE1, Peer IP: 192.0.2.1
  Source IP: 192.0.2.2, Total routes: 1
    Flags  Prefix                              Next Hop               MED
Lpref       AS Path
    !      192.168.2.11/32                     192.0.2.1              0           -
65536
Instance: default, AFI: ipv4, SAFI: unicast
  Hostname: PE1, Peer IP: 192::1
  Source IP: 192::2, Total routes: 1
    Flags  Prefix                              Next Hop               MED
Lpref       AS Path
    !      192.168.2.11/32                     192::1                 0           -
65536
```

Example 10: Summary view of the BGP rib-in post-policy for a specific instance, AFI, SAFI, and peer address filter.

```
supervisor@rtbrick.net: op> show bgp rib-in post-policy ipv4 unicast peer address
192.0.2.1
Flags: & - Imported, ! - Error, N - RPKI Unknown, I - RPKI Invalid, V - RPKI Valid
Instance: default, AFI: ipv4, SAFI: unicast
  Hostname: PE1, Peer IP: 192.0.2.1
  Source IP: 192.0.2.2, Total routes: 3
    Flags  Prefix                             Next Hop               MED         Lpref
AS Path
           192.0.2.10/32                      192.0.2.1              0           -
65536
           192.0.2.20/32                      192.0.2.1              0           -
65536
           192.0.2.11/32                      192.0.2.1              0           -
65536
```

Example 11: Summary view of the BGP rib-in post-policy for a specific instance, AFI, SAFI, and validation-state filter.

```
supervisor@rtbrick.net: op> show bgp rib-in post-policy ipv4 unicast validation-state
init
Flags: & - Imported, ! - Error, N - RPKI Unknown, I - RPKI Invalid, V - RPKI Valid
Instance: default, AFI: ipv4, SAFI: unicast
  Hostname: PE1, Peer IP: 192.0.2.1
  Source IP: 192.0.2.2
    Flags  Prefix                                Next Hop           MED         Lpref
```

```
AS Path
           192.0.2.10/32                          192.0.2.1          0           -
65536
           192.0.2.20/32                          192.0.2.1          0           -
65536
           192.0.2.11/32                          192.0.2.1          0           -
65536
Instance: default, AFI: ipv4, SAFI: unicast
  Hostname: PE1, Peer IP: 192::1
  Source IP: 192::2
    Flags  Prefix                                 Next Hop           MED         Lpref
AS Path
           192.0.2.10/32                          192::1             0           -
65536
           192.0.2.20/32                          192::1             0           -
65536
           192.0.2.11/32                          192::1             0           -
65536
```

#### BGP RIB-out

This command displays the send routes.

**Syntax:**

**show bgp rib-out** <option> ...

| Option | Description |
|---|---|
| - | Without any option, the command displays advertised BGP routes for all instances. |
| <afi> | BGP routing table summary for the given address family (AFI), all sub-address families and all instances. Supported AFI values are 'ipv4' and 'ipv6'. |
| <afi> <safi> | BGP routing table summary for the given address family (AFI) and sub-address family (SAFI), and all instances. Supported SAFI values are evpn, evpn-vpws, flowspec, labeled-unicast, multicast, unicast, vpls, vpls-vpws, vpn-multicast, and vpn-unicast. |
| <afi> <safi> detail | Detailed list view of the BGP routing table for the given address family (AFI) and sub-address family (SAFI), and all instances. |
| <afi> <safi> <prefix> | BGP routing table entry for the given prefix and all instances. |
| <afi> <safi> instance <instance-name> | BGP routing table summary for the given AFI, SAFI, and instance. |
| <afi> <safi> instance <instance-name> detail | Detailed list view of BGP routing table for the given AFI, SAFI, and instance. |

| Option | Description |
|---|---|
| `<afi> <safi> peer address <peer-ip>` | View of BGP routing table for the given peer Ip address. |
| `<afi> <safi> peer address <peer-ip> detail` | Detailed list view of BGP routes for give peer IP address. |
| `<afi> <safi> peer <peer-name>` | Displays BGP routing table for the given peer. |
| `<afi> <safi> peer <peer-name>` | Detailed list view of BGP routes for the given peer IP address. |
| `<afi> <safi> peer-group <name>` | Information of BGP routes for the given peer group name. |
| `<afi> <safi> community <community-name>` | Displays BGP routing table for the given community. |
| `<afi> <safi> community <community-name> detail` | Detailed list view of BGP routing table for the given community. |
| `<afi> <safi> instance <instance-name> <prefix>` | BGP routing table entry for the given prefix and instance. |
| `<afi> <safi> peer <name> / peer address <ip>` | Peer name or address |

Example 1: Summary view of the routes advertised to a peer

```
supervisor@rtbrick: op> show bgp rib-out ipv4 unicast peer PE2
Flags: N - RPKI Unknown, I - RPKI Invalid, V - RPKI Valid
Instance: default, AFI: ipv4, SAFI: unicast
  Peer: PE2, Peer IP: 192.0.2.2, Sent routes: 8
    Flags  Prefix                                        MED     Lpref      Origin
Next Hop                                AS Path
           192.0.2.0/30                                  0       -          Incomplete
192.0.2.1                               65541
           192.0.2.4/30                                  0       -          Incomplete
192.0.2.1                               65541
           198.51.100.11/32                              0       -          Incomplete
192.0.2.1                               65541
           198.51.100.12/32                              0       -          Incomplete
192.0.2.1                               65541
           198.51.100.21/32                              0       -          Incomplete
192.0.2.1                               65541, 65542
           198.51.100.22/32                              0       -          Incomplete
192.0.2.1                               65541, 65542
           198.51.100.100/32                             0       -          Incomplete
192.0.2.1                               65541
           198.51.100.101/32                             0       -          Incomplete
192.0.2.1                               65541
```

Example 2: Detailed view of the routes advertised to a peer

```
supervisor@rtbrick: op> show bgp rib-out
Flags: N - RPKI Unknown, I - RPKI Invalid, V - RPKI Valid
```

```
Instance: default, AFI: ipv4, SAFI: unicast
  Peer-group: PE2, Sent routes: 8
    Flags  Prefix                                        MED     Lpref      Origin
Next Hop                                 AS Path
           192.0.2.0/30                                  0       -          Incomplete
-                                        65541
           192.0.2.4/30                                  0       -          Incomplete
-                                        65541
           198.51.100.11/32                              0       -          Incomplete
-                                        65541
           198.51.100.12/32                              0       -          Incomplete
-                                        65541
           198.51.100.21/32                              0       -          Incomplete
-                                        65541, 65542
           198.51.100.22/32                              0       -          Incomplete
-                                        65541, 65542
           198.51.100.100/32                             0       -          Incomplete
-                                        65541
           198.51.100.101/32                             0       -          Incomplete
-                                        65541
Instance: default, AFI: ipv4, SAFI: labeled-unicast
  Peer-group: PE2, Sent routes: 6
    Flags  Prefix                                        MED     Lpref      Origin
Next Hop                                 AS Path
           192.0.2.0/30                                  0       -          Incomplete
-                                        65541
           192.0.2.4/30                                  0       -          Incomplete
-                                        65541
           198.51.100.11/32                              0       -          Incomplete
-                                        65541
           198.51.100.12/32                              0       -          Incomplete
-                                        65541
           198.51.100.21/32                              0       -          Incomplete
-                                        65541, 65542
           198.51.100.22/32                              0       -          Incomplete
-                                        65541, 65542
Instance: default, AFI: ipv6, SAFI: unicast
  Peer-group: PE2, Sent routes: 6
    Flags  Prefix                                        MED     Lpref      Origin
Next Hop                                 AS Path
           2001:db8:1:1::/64                             0       -          Incomplete
-                                        65541
           2001:db8:2:1::/64                             0       -          Incomplete
-                                        65541
           2001:db8:0:1::11/128                          0       -          Incomplete
-                                        65541
           2001:db8:0:1::12/128                          0       -          Incomplete
-                                        65541
           2001:db8:0:1::21/128                          0       -          Incomplete
-                                        65541, 65542
           2001:db8:0:1::22/128                          0       -          Incomplete
```

```
-                                       65541, 65542
Instance: default, AFI: ipv6, SAFI: labeled-unicast
  Peer-group: PE2, Sent routes: 6
    Flags  Prefix                                       MED     Lpref      Origin
Next Hop                                AS Path
           2001:db8:1:1::/64                            0       -          Incomplete
-                                       65541
           2001:db8:2:1::/64                            0       -          Incomplete
-                                       65541
           2001:db8:0:1::11/128                         0       -          Incomplete
-                                       65541
           2001:db8:0:1::12/128                         0       -          Incomplete
-                                       65541
           2001:db8:0:1::21/128                         0       -          Incomplete
-                                       65541, 65542
           2001:db8:0:1::22/128                         0       -          Incomplete
-                                       65541, 65542
```

#### TCP Connections

This command displays information of the TCP connections used by BGP.

**Syntax:**

**show bgp tcp bgp.iod.1 connection** <option> ...

| Option | Description |
|---|---|
| - | Without any option, the command displays the TCP connections used by BGP for all instances. |
| detail | Detailed list view of the the TCP connections for the given address family (AFI) and sub-address family (SAFI), and all instances. |
| prefix | TCP connections for the given prefix and all instances. |
| instance <instance-name> | TCP connections summary for the given instance. |

Example 1: Summary view of the BGP TCP connections

```
supervisor@rtbrick: op> show bgp rib-out ipv4 unicast peer PE2
Flags: N - RPKI Unknown, I - RPKI Invalid, V - RPKI Valid
Instance: default, AFI: ipv4, SAFI: unicast
  Peer: PE2, Peer IP: 192.0.2.2, Sent routes: 8
    Flags  Prefix                                       MED     Lpref      Origin
Next Hop                                AS Path
           192.0.2.0/30                                 0       -          Incomplete
192.0.2.1                               65541
           192.0.2.4/30                                 0       -          Incomplete
```

```
192.0.2.1                               65541
           198.51.100.11/32                             0       -          Incomplete
192.0.2.1                               65541
           198.51.100.12/32                             0       -          Incomplete
192.0.2.1                               65541
           198.51.100.21/32                             0       -          Incomplete
192.0.2.1                               65541, 65542
           198.51.100.22/32                             0       -          Incomplete
192.0.2.1                               65541, 65542
           198.51.100.100/32                            0       -          Incomplete
192.0.2.1                               65541
           198.51.100.101/32                            0       -          Incomplete
192.0.2.1                               65541
```

```
supervisor@rtbrick: op> show bgp tcp bgp.iod.1 connection
Instance       Local IP Address                  Remote IP Address
Local Port  Remote Port  State
default        192.0.2.1                         192.0.2.2
179         58311        Time-Wait
default        192.0.2.1                         192.0.2.2
179         58312        Time-Wait
default        192.0.2.1                         192.0.2.2
58312       179          Time-Wait
default        192.0.2.1                         192.0.2.2
58313       179          Established
```

Example 2: Detailed information of the BGP TCP connections

```
supervisor@rtbrick: op> show bgp tcp bgp.iod.1 connection detail
Instance: default
  Local IPv4 address       : 192.0.2.1
  Remote IPv4 address      : 192.0.2.2
  Local port               : 179
  Remote port              : 58311
  State                    :
    Internal
      Options                  : -- | Keepalive | --
      TOS                      : 192
      TTL                      : 1
      Priority                 : 1
      Flags                    : -|-|-|Receive Closed|Fin|Nagle Disabled|-|Wnd Scale|-
|-|-
      Last trigger             : 148
      Timer                    : 29
    Timers
      Poll                     : 0s
      Poll interval            : 0s
      Retransmission           : 65535s
```

```
    Receiver
      Expected sequence        : 7156
      Available window         : 96816
      Announced window         : 96185
      Announced wnd RT edge    : 103341
      MSS                      : 1420
      RTT estimate             : 0
    Timeout
      Sequence                 : 7193s
      Retransmission           : 3s
      Retransmissions          : 0s
      Duplicate acks           : 0s
      Highest ack'd sequence   : 7194s
    Congestion
      Window                   : 5042
      Persist count            : 0
      Send scale               : 5
      Receive scale            : 5
    Sender
      Next seq to send         : 7194
      Last wnd update seq      : 7155
      Last wnd update ack      : 7194
      Window                   : 96128
      Max window announced     : 96800
      Acknowledged             : None
      Send buf                 : 56476
      Send queue length        : 0
      Unsent oversize          : 0
      TS last ack sent         : 0
    Keepalive
      Next keepalive idle      : 7200000
      Keepalive interval       : 75000
      Keepalive count          : 9
      Keep sent count          : 0
    Authentication
      Auth type                : HMAC-SHA-256-128
      key1-id                  : 255
      key2-id                  : 0
      Secret mismatch          : 0
      Latest sent digest       : 13ff781f8b9a7980f0bc6aeed5d00800
      Latest received digest   : ced8b4c0d6ec7772781d3751e9ad4d6c
    TTL Security
      TTL security             : False
      Expected TTL             : 0
      TTL miss count           : 0
  Local IPv4 address       : 192.0.2.1
  Remote IPv4 address      : 192.0.2.2
  Local port               : 179
  Remote port              : 58312
  State                    :
    Internal
```

```
      Options                  : -- | Keepalive | --
      TOS                      : 192
      TTL                      : 1
      Priority                 : 1
      Flags                    : -|-|-|Receive Closed|Fin|Nagle Disabled|-|Wnd Scale|-
|-|-
      Last trigger             : 159
      Timer                    : 31
    Timers
      Poll                     : 0s
      Poll interval            : 0s
      Retransmission           : 65535s
    Receiver
      Expected sequence        : 6673
      Available window         : 96816
      Announced window         : 96724
      Announced wnd RT edge    : 103397
      MSS                      : 1420
      RTT estimate             : 0
    Timeout
      Sequence                 : 6594s
      Retransmission           : 5s
      Retransmissions          : 0s
      Duplicate acks           : 0s
      Highest ack'd sequence   : 6595s
    Congestion
      Window                   : 4381
      Persist count            : 0
      Send scale               : 5
      Receive scale            : 5
    Sender
      Next seq to send         : 6595
      Last wnd update seq      : 6672
      Last wnd update ack      : 6595
      Window                   : 96800
      Max window announced     : 96800
      Acknowledged             : None
      Send buf                 : 56476
      Send queue length        : 0
      Unsent oversize          : 0
      TS last ack sent         : 0
    Keepalive
      Next keepalive idle      : 7200000
      Keepalive interval       : 75000
      Keepalive count          : 9
      Keep sent count          : 0
    Authentication
      Auth type                : HMAC-SHA-256-128
      key1-id                  : 255
      key2-id                  : 0
      Secret mismatch          : 0
```

```
      Latest sent digest       : 8c63c66ce0b6db0ac0a52481ce1be90c
      Latest received digest   : da1923f15a39ce953bcfd3c90bfd1d71
    TTL Security
      TTL security             : False
      Expected TTL             : 0
      TTL miss count           : 0
  Local IPv4 address       : 192.0.2.1
  Remote IPv4 address      : 192.0.2.2
  Local port               : 58312
  Remote port              : 179
  State                    :
    Internal
      Options                  : -- | Keepalive | --
      TOS                      : 192
      TTL                      : 1
      Priority                 : 64
      Flags                    : -|-|-|Receive Closed|Fin|Nagle Disabled|-|Wnd Scale|-
|-|-
      Last trigger             : 159
      Timer                    : 31
    Timers
      Poll                     : 0s
      Poll interval            : 0s
      Retransmission           : 65535s
    Receiver
      Expected sequence        : 6576
      Available window         : 96816
      Announced window         : 96794
      Announced wnd RT edge    : 103370
      MSS                      : 1420
      RTT estimate             : 0
    Timeout
      Sequence                 : 6653s
      Retransmission           : 5s
      Retransmissions          : 0s
      Duplicate acks           : 0s
      Highest ack'd sequence   : 6654s
    Congestion
      Window                   : 4472
      Persist count            : 0
      Send scale               : 5
      Receive scale            : 5
    Sender
      Next seq to send         : 6654
      Last wnd update seq      : 6575
      Last wnd update ack      : 6654
      Window                   : 96704
      Max window announced     : 96704
      Acknowledged             : None
      Send buf                 : 56476
      Send queue length        : 0
```

```
      Unsent oversize          : 0
      TS last ack sent         : 0
    Keepalive
      Next keepalive idle      : 7200000
      Keepalive interval       : 75000
      Keepalive count          : 9
      Keep sent count          : 0
    Authentication
      Auth type                : HMAC-SHA-256-128
      key1-id                  : 255
      key2-id                  : 0
      Secret mismatch          : 0
      Latest sent digest       : 3fe0662305214b8bf9fafa07a1594a25
      Latest received digest   : 7aa39a7ff33bffcc0ad74e8e98ee6ce1
    TTL Security
      TTL security             : False
      Expected TTL             : 0
      TTL miss count           : 0
  Local IPv4 address       : 192.0.2.1
  Remote IPv4 address      : 192.0.2.2
  Local port               : 58313
  Remote port              : 179
  State                    : Established
    Internal
      Options                  : -- | Keepalive | --
      TOS                      : 192
      TTL                      : 1
      Priority                 : 64
      Flags                    : -|-|-|-|-|Nagle Disabled|-|Wnd Scale|-|-|-
      Last trigger             : 142
      Timer                    : 73
    Timers
      Poll                     : 0s
      Poll interval            : 0s
      Retransmission           : 65535s
    Receiver
      Expected sequence        : 8177
      Available window         : 96816
      Announced window         : 96816
      Announced wnd RT edge    : 104993
      MSS                      : 1420
      RTT estimate             : 0
    Timeout
      Sequence                 : 7270s
      Retransmission           : 3s
      Retransmissions          : 0s
      Duplicate acks           : 0s
      Highest ack'd sequence   : 7326s
    Congestion
      Window                   : 5076
      Persist count            : 0
```

```
      Send scale                : 5
      Receive scale             : 5
    Sender
      Next seq to send          : 7326
      Last wnd update seq       : 7721
      Last wnd update ack       : 7326
      Window                    : 96096
      Max window announced      : 96704
      Acknowledged              : None
      Send buf                  : 56476
      Send queue length         : 0
      Unsent oversize           : 0
      TS last ack sent          : 0
    Keepalive
      Next keepalive idle       : 7200000
      Keepalive interval        : 75000
      Keepalive count           : 9
      Keep sent count           : 0
    Authentication
      Auth type                 : HMAC-SHA-256-128
      key1-id                   : 255
      key2-id                   : 0
      Secret mismatch           : 0
      Latest sent digest        : b3b62fbef0b644235ee047713e3e0565
      Latest received digest    : 826ce5c244bb8f227bbf088e61d53cd4
    TTL Security
      TTL security              : False
      Expected TTL              : 0
      TTL miss count            : 0
```

#### TCP Statistics

This command displays TCP statistics information of the TCP connections used by BGP.

**Syntax:**

**show bgp tcp bgp.iod.1 statistics** <option> ...

| Option | Description |
|---|---|
| - | Without any option, the command displays the TCP statistics information of the TCP connections used by BGP for all instances. |
| instance <instance-name> | TCP connections summary for the given instance. |

Example: TCP statistics information of the TCP connections used by BGP for the default instance

```
supervisor@rtbrick: op> show bgp tcp bgp.iod.1 statistics instance default
Instance: default
```

```
  IP statistics
    Transmitted packets    : 3103242412
    Received packets       : 47351
    Forwarded packets      : 0
    Dropped packets        : 0
    Checksum error         : 0
    Invalid length error   : 0
    Out of memory error    : 0
    Routing error          : 0
    Protocol error         : 0
    Error in options       : 0
    Misc error             : 0
    Cachehit               : 0
  TCP statistics
    Transmitted packets    : 365499779
    Received packets       : 5577
    Forwarded packets      : 3014656
    Dropped packets        : 46
    Checksum error         : 0
    Invalid length error   : 0
    Out of memory error    : 0
    Routing error          : 3014656
    Protocol error         : 46
    Error in options       : 0
    Misc error             : 2097152
    Cachehit               : 1557594144
```

### BGP Clear Commands

Clear commands allow to reset operational states.

#### BGP Peer

This commands resets BGP peerings.

**Syntax:**

**clear bgp peer** <option> ...

| Option | Description |
|---|---|
| `all` | Clears all the BGP peers. |
| `all soft-in <afi> <safi>` | Sends route refresh to all neighbors. |
| `all soft-out <afi> <safi>` | Re-advertises all the routes previously sent to the peer. |
| `all stats` | Clears the statistics of all the BGP peers. |
| `instance <instance> <peer-ip>` | Clears the peer for the given instance and peer IP address. |

| Option | Description |
|---|---|
| `instance <instance> <peer-ip> source <src-ip>` | Clears a specific peer for the given peer IP address and source IP address in the specified instance. |
| `instance <instance> all` | Clears all peers in the given instance. |
| `instance <instance> <peer-ip> source <src-ip> soft-in <afi> <safi>` | Sends route refresh to specific peer for the given instance, peer-ip, source-ip and address-family. |
| `instance <instance> <peer-ip> soft-in <afi> <safi>` | Sends route refresh to peer for the given instance, peer-ip and address-family. |
| `instance <instance> all soft-in <afi> <safi>` | Sends route refresh to all peers for the given instance and address family. |
| `instance <instance> <peer-ip> source <src-ip> soft-out <afi> <safi>` | Re-advertises all the routes previously sent to the specific peer for the given instance, peer-ip, source-ip and address-family. |
| `instance <instance> <peer-ip> soft-out <afi> <safi>` | Sends route refresh to peer for a given instance, peer-ip and address-family. |
| `instance <instance> all soft-out <afi> <safi>` | Sends route update to all peers for given instance and address family. |
| `instance <instance> <peer-ip> source <src-ip> stats` | Clears the statistics of a specific peer for a given instance, peer-ip and source-IP. |
| `instance <instance> <peer-ip> stats` | Clears the statistics of the peer for a given instance and peer-IP. |
| `instance <instance> all stats` | Clears the statistics of all peers for a given instance. |

Example: The example below shows how to clear all the BGP peers.

```
supervisor@rtbrick: op> clear bgp peer all
```

API To access the Operational State API that corresponds to this CLI, click here.

# 2.4. BGP FlowSpec

## 2.4.1. BGP FlowSpec Overview

BGP FlowSpec is an extension of the BGP protocol that allows for the dynamic propagation of more specific information than the traffic aggregate defined by an IP Prefix. This enables network administrators to control data traffic flow at any point in their network infrastructure. BGP FlowSpec can be used for various purposes, such as managing congestion or mitigating distributed denial-of-service (DDoS) attacks. Expanding routing information with FlowSpec allows the routing system to use the ACL (Access Control List) or firewall capabilities in the router's forwarding path.

The figure below shows a traffic scrubbing station capable of generating the FlowSpec rule in the event of a DDoS attack and sending the BGP FlowSpec update to the neighbouring devices. The device that can interpret this update can drop the DDoS traffic as it arrives.

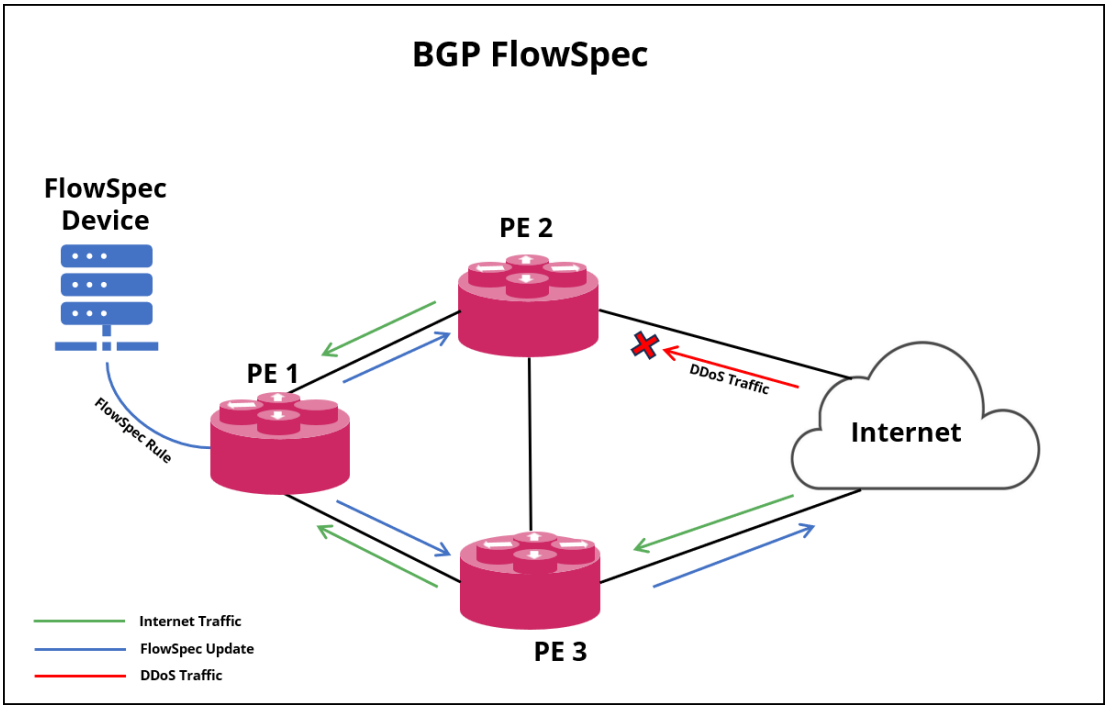



*Figure 8. BGP FlowSpec Overview*

FlowSpec helps to establish matching criteria for IP traffic packets encoded into BGP Network Layer Reachability Information (NLRI). The requirements can include various attributes and may or may not involve reachability information. Routers can use FlowSpec to forward, shape, classify, rate limit, filter, or redirect packets based on specific policies, allowing for rules that operate on multiple fields of the packet header.

The figure below shows the high-level flow of BGP FlowSpec design.

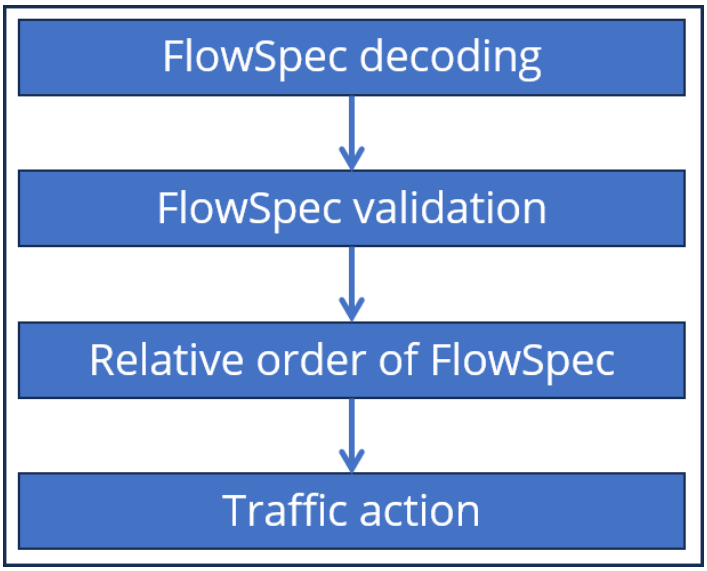



*Figure 9. High-level flow of BGP FlowSpec design*

FlowSpec involves actively processing and inserting dynamic ACLs in an operational environment. When a router receives a FlowSpec update, it can dynamically create IP filters to mitigate intra-AS and inter-AS DDoS attacks and other unwanted traffic patterns. Mitigation is implemented by dropping or rate-limiting the traffic at the network's ingress point (or the nearest possible point toward the source of the DDoS attack).

Matching of TCP flags is enabled in BGP FlowSpec.

### Supported BGP Standards

| RFC Number | Description |
|---|---|
| RFC 8955 | Dissemination of Flow Specification Rules for IPv4 |
| RFC 8956 | Dissemination of Flow Specification Rules for IPv6 |

RFC and draft compliance are partial except as specified.

### Supported Platforms

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

### Supported Matching Criteria and Actions

The tables below outline the FlowSpec supported matching criteria and actions.

| BGP FlowSpec NLRI Type | Match Criteria & Description | | Option | Supported (Yes/No) |
|---|---|---|---|---|
| Type 1 | Destination-Prefix (ipv4/ipv6) | Defines the destination prefix to match | specific host | Yes |
| | | | IP range | Yes |
| Type 2 | Source-Prefix (ipv4/ipv6) | Defines the source prefix to match. | specific host | Yes |
| | | | IP range | Yes |
| Type 3 | IP-Protocol | Contains a set of {operator, value} pairs that match the IP protocol value byte in IP packets. | specific value | Yes |
| | | | multi-value range | No |
| Type 4 | Port (src or dst) | Defines a list of {operator, value} pairs that match source or destination ports. | specific value | No |
| | | | multi-value range | No |
| Type 5 | Destination port | Defines a list of {operator, value} pairs used to match the destination port of a TCP or UDP packet. | specific value | Yes |
| | | | multi-value range | No |

| BGP FlowSpec NLRI Type | Match Criteria & Description | | Option | Supported (Yes/No) |
|---|---|---|---|---|
| Type 6 | Source port | Defines a list of {operator, value} pairs used to match the source port of a TCP or UDP packet. | specific value | Yes |
| | | | multi-value range | No |
| Type 7 | ICMP type | Defines a list of {operator, value} pairs used to match the type field of an ICMP packet | specific value | No |
| | | | multi-value range | No |
| Type 8 | ICMP code | Defines a list of {operator, value} pairs used to match the code field of an ICMP packet. | specific value | No |
| | | | multi-value range | No |
| Type 9 | TCP flag | IPv4 or IPv6 TCP flags(2 bytes include reserved bits) | specific value | Yes |
| | | | multi-value range | Yes |
| Type 10 | Packet length | Match on the total IP packet length | specific value | No |
| | | | multi-value range | No |
| Type 11 | DSCP | Defines a list of {operator, value} pairs that use a Multi-value range to match the 6-bit DSCP field. | specific value | No |
| | | | multi-value range | No |
| Type 12 | Fragment | Identifies a fragment-type as the match Bit mask criterion for a class map. | specific value | No |
| | | | multi-value range | No |

The maximum number of matches supported in a single FlowSpec rule is 8.

The table below outlines the supported actions for BGP FlowSpec rules.

| | Action Criteria & Description | | Option | Supported (Yes/No) |
|---|---|---|---|---|
| 1 | traffic-rate-bytes | Traffic-rate limits specified in bytes per second | 0 (drop) | Yes |
| | | | >0 | Yes |
| 2 | traffic-rate-packets | Traffic-rate limits specified in packets per second | 0 (drop) | Yes |
| | | | >0 | No |
| 3 | traffic-action | Action that is performed on the traffic that matches FlowSpec rule | Terminal | No |
| | | | Sampling | No |
| 4 | rt-redirect | Redirects the traffic to a specific VRF instance or to a next-hop | to vrf | No |
| | | | to nexthop | No |

The user-defined ACL priority range is 100 to 20000, while BGP FlowSpec priority ranges from 1501 to 2500, which prevents conflicts during configuration.

**Operators Supported in Matching Criteria and Actions**

| Operator Type | Supported Operators | Unsupported Operators |
|---|---|---|
| Relational | `equal` | `greater than`, `greater than or equal`, `less than`, `less than or equal`, `not equal` |
| Logical | `or` | `and` |

## 2.4.2. BGP FlowSpec Configuration

### Enabling BGP FlowSpec Address Family

Adding the FlowSpec address family to the BGP address-family configuration enables the exchange of BGP FlowSpec NLRIs.

**Syntax:**

```
set <instance> <instance-name> protocol bgp address-family <afi> <safi>
```

| Attribute | Description |
|---|---|
| <instance-name> | Name of the BGP instance |
| <afi> | Address family identifier (AFI). Supported values: ipv4, ipv6 |
| <safi> | Subsequent address family identifier (SAFI). Specify "flowspec" as the SAFI for BGP to enable the exchange of BGP FlowSpec NLRIs. |

Example 1: BGP FlowSpec IPv4 Configuration

```
supervisor@rtbrick>LEAF01: cfg> show config instance default protocol bgp address-
family ipv4 flowspec
{
  "rtbrick-config:address-family": [
    {
      "afi": "ipv4",
      "safi": "flowspec"
    }
  ]
}
```

Example 2: BGP FlowSpec IPv6 Configuration

```
supervisor@rtbrick>LEAF01: cfg> show config instance default protocol bgp address-
family ipv6 flowspec
{
  "rtbrick-config:address-family": [
    {
      "afi": "ipv6",
      "safi": "flowspec"
    }
  ]
}
```

**Enabling BGP FlowSpec in Peer Group Address Family**

**Syntax:**

**set instance** <instance-name> **protocol bgp peer-group** <peer-group-name> **address-family** <afi> <safi> <attribute> <value>

| Attribute | Description |
|---|---|
| <afi> | Address family identifier (AFI). Supported values: ipv4, ipv6, l2vpn |

| Attribute | Description |
| --- | --- |
| <safi> | Subsequent address family identifier (SAFI). Specify "flowspec" as the SAFI to enable BGP FlowSpec in Peer Group address family. |

Example 1: BGP FlowSpec in Peer Group Address Family

```
supervisor@rtbrick>LEAF01: cfg> show config instance default protocol bgp peer-group
{
  "rtbrick-config:peer-group": [
    {
      "pg-name": "PE2",
      "remote-as": 4200000002,
      "address-family": [
        {
          "afi": "ipv4",
          "safi": "flowspec"
        },
        {
          "afi": "ipv4",
          "safi": "unicast"
        },
        {
          "afi": "ipv6",
          "safi": "flowspec"
        },
        {
          "afi": "ipv6",
          "safi": "unicast"
        }
      ]
    }
  ]
}
```

## 2.4.3. BGP FlowSpec Operational Commands

### BGP Show Commands

#### show bgp peer

The 'show bgp peer' commands display information on BGP peers.

**Syntax:**

**show bgp peer** <option> ...

| Option | Description |
|---|---|
| - | Without any option, the commands display all BGP peers in all instances in a summary table format. |
| detail | Detailed information on all BGP peers in all instances in a list view. |
| <peer-name> | Detailed information on the peer with the given name. |
| history | Displays BGP peer history information such as the peer state down reasons. |
| history <peer-address> | Displays BGP peer history information such as the peer state down reasons for a specified peer. |
| address <peer-address> | Detailed information on the peer with the given IP address. |
| instance <instance-name> | Summary of all BGP peers in the given instance. |
| instance <instance-name> detail | Detailed information on all BGP peers in the given instance. |
| instance <instance-name> detail <peer-name> | Detailed information on the peer with the given name in the given instance. |
| instance <instance-name> detail address <peer-address> | Detailed information on the peer with the given IP address in the given instance. |
| statistics | Received and sent BGP prefixes per AFI/SAFI for all peers in all instances. |
| statistics peer <peer-name> | Received and sent BGP prefixes per AFI/SAFI for the peer with the given name. |
| statistics peer address <peer-address> | Received and sent BGP prefixes per AFI/SAFI for the peer with the given IP address. |
| statistics instance <instance-name> peer <peer-name> | Received and sent BGP prefixes per AFI/SAFI for the peer with the given name in the given instance. |
| statistics instance <instance-name> peer address <peer-address> | Received and sent BGP prefixes per AFI/SAFI for the peer with the given IP address in the given instance |

Example 1: BGP Peer Summary View

```
supervisor@rtbrick: op> show bgp peer
Instance: default
  Peer                                    Remote AS    State         Up/Down Time               PfxRcvd
PfxSent
  PE2                                     65542        Established   0d:00h:00m:21s             8                    17
  SN-STD-2-3966602                        65549        Established   0d:00h:00m:23s             3                    3
```

Example 2: BGP Peer Detail View

```
supervisor@rtbrick: op> show bgp peer detail
Peer: PE2, Peer IP: 192.0.2.2, Remote AS: 65542, Local IP: 192.0.2.1, Local AS: 65541, Any AS: False
  Type: ebgp, State: Established, Up/Down Time: Wed Jul 15 19:25:47 GMT +0000 2026
```

```
  Discovered on interface: -
  Last transition: Wed Jul 15 19:25:47 GMT +0000 2026, Flap count: 0
  Peer ID        : 198.51.100.21, Local ID: 198.51.100.11
  Instance       : default, Peer group: PE2
  6PE enabled    : False
  TTL security   : False, TTL limit: -
  Deactivate     : False
  Authentication:
    ID: BGP_AUTH_HMAC-SHA-256-128, Algorithm: HMAC-SHA-256-128
  Timer values:
    Peer keepalive : 30s, Local keepalive: 30s
    Peer holddown  : 90s, Local holddown : 90s
    Connect retry  : 30s
  Timers:
    Connect retry timer : 0
    Keepalive timer     : expires in 6s 536415us
    Holddown timer      : expires in 1m 8s 418579us
  NLRIs:
    Sent           : ['ipv4-unicast', 'ipv6-unicast', 'ipv4-flowspec', 'ipv6-flowspec', 'ipv4-labeled-unicast', 'ipv6-
labeled-unicast']
    Received       : ['ipv4-unicast', 'ipv6-unicast', 'ipv4-flowspec', 'ipv6-flowspec', 'ipv4-labeled-unicast', 'ipv6-
labeled-unicast']
    Negotiated     : ['ipv4-unicast', 'ipv6-unicast', 'ipv4-flowspec', 'ipv6-flowspec', 'ipv4-labeled-unicast', 'ipv6-
labeled-unicast']
  Capabilities:
    Addpath sent                  : None
    Addpath received              : None
    Addpath negotiated            : None
    Extended nexthop sent         : None
    Extended nexthop received     : None
    Extended nexthop negotiated   : None
    Multiple labels sent          : None
    Multiple labels received      : None
    Multiple labels negotiated    : None
    Capabilities:
      Feature                      Sent            Received        Negotiated
      Route refresh                True            True            True
      4 byte AS                    True            True            True
      Graceful restart             False           False           False
      Link local only              False           False           False
  Prefix Limit:
  End of RIB:
    Address family                 Sent                                Received
    IPv4 unicast                   Wed Jul 15 19:25:53 GMT +0000 2026  Wed Jul 15 19:25:53 GMT +0000 2026
    IPv4 labeled-unicast           Wed Jul 15 19:25:53 GMT +0000 2026  Wed Jul 15 19:25:53 GMT +0000 2026
    IPv6 unicast                   Wed Jul 15 19:25:53 GMT +0000 2026  Wed Jul 15 19:25:53 GMT +0000 2026
    IPv6 labeled-unicast           Wed Jul 15 19:25:53 GMT +0000 2026  Wed Jul 15 19:25:53 GMT +0000 2026
    IPv4 VPN-unicast               never                               never
    IPv6 VPN-unicast               never                               never
    IPv4 flowspec                  Wed Jul 15 19:25:53 GMT +0000 2026  Wed Jul 15 19:25:53 GMT +0000 2026
    IPv6 flowspec                  Wed Jul 15 19:25:53 GMT +0000 2026  Wed Jul 15 19:25:53 GMT +0000 2026
    IPv4 VPN-multicast             never                               never
    L2VPN VPLS                     never                               never
    L2VPN EVPN                     never                               never
  RPKI Validation Stats:
    Address family                 Valid    Invalid    Unknown
    IPv4 unicast                       0          0          0
    IPv6 unicast                       0          0          0
  Message stats:
    Session stats:
      Direction              Open          Update       Keepalive          Notify   Route refresh
      Input                     1               8               1               0               0
      Output                    1              11               1               0               0
    Total stats:
      Input                     1               8               1               0               0
      Output                    1              11               1               0               0
```

```
    Route stats:
      Address family                        Received            Sent    Prefix limit    Idle timeout
      IPv4 unicast                                 4               7               0               0
      IPv4 labeled-unicast                         0               0               0               0
      IPv6 unicast                                 4               7               0               0
      IPv6 labeled-unicast                         0               0               0               0
      IPv4 VPN-unicast                             0               0               0               0
      IPv6 VPN-unicast                             0               0               0               0
      IPv4 multicast                               0               0               0               0
      IPv4 VPN-multicast                           0               0               0               0
      L2VPN VPLS                                   0               0               0               0
      L2VPN EVPN                                   0               0               0               0
      IPv4 Flowspec                                0               2               0               0
      IPv6 Flowspec                                0               1               0               0
Peer: SN-STD-2-3966602, Peer IP: 192.0.2.102, Remote AS: 65549, Local IP: 192.0.2.101, Local AS: 65541, Any AS: False
  Type: ebgp, State: Established, Up/Down Time: Wed Jul 15 19:25:45 GMT +0000 2026
  Discovered on interface: -
  Last transition: Wed Jul 15 19:25:45 GMT +0000 2026, Flap count: 0
  Peer ID        : 198.51.100.91, Local ID: 198.51.100.11
  Instance       : default, Peer group: FS
  6PE enabled    : False
  TTL security   : False, TTL limit: -
  Authentication:
    ID: , Algorithm: HMAC-SHA-256-128
  Timer values:
    Peer keepalive : 30s, Local keepalive: 30s
    Peer holddown  : 90s, Local holddown : 90s
    Connect retry  : 30s
  Timers:
    Connect retry timer : 0
    Keepalive timer     : expires in 2s 632207us
    Holddown timer      : expires in 1m 6s 383572us
  NLRIs:
    Sent           : ['ipv4-flowspec', 'ipv6-flowspec']
    Received       : ['ipv4-flowspec', 'ipv6-flowspec']
    Negotiated     : ['ipv4-flowspec', 'ipv6-flowspec']
  Capabilities:
    Addpath sent                  : None
    Addpath received              : None
    Addpath negotiated            : None
    Extended nexthop sent         : None
    Extended nexthop received     : ['ipv4-flowspec', 'ipv6-flowspec']
    Extended nexthop negotiated   : None
    Multiple labels sent          : None
    Multiple labels received      : None
    Multiple labels negotiated    : None
    Capabilities:
      Feature                      Sent            Received        Negotiated
      Route refresh                True            True            True
      4 byte AS                    True            True            True
      Graceful restart             False           False           False
      Link local only              False           False           False
  Prefix Limit:
  End of RIB:
    Address family                 Sent                                Received
    IPv4 unicast                   never                               never
    IPv4 labeled-unicast           never                               never
    IPv6 unicast                   never                               never
    IPv6 labeled-unicast           never                               never
    IPv4 VPN-unicast               never                               never
    IPv6 VPN-unicast               never                               never
    IPv4 flowspec                  Wed Jul 15 19:25:51 GMT +0000 2026  Wed Jul 15 19:25:51 GMT +0000 2026
    IPv6 flowspec                  Wed Jul 15 19:25:51 GMT +0000 2026  Wed Jul 15 19:25:51 GMT +0000 2026
    IPv4 VPN-multicast             never                               never
    L2VPN VPLS                     never                               never
    L2VPN EVPN                     never                               never
```

```
  Message stats:
    Session stats:
      Direction              Open          Update       Keepalive          Notify   Route refresh
      Input                     1               3               1               0               0
      Output                    1               2               1               0               0
    Total stats:
      Input                     1               3               1               0               0
      Output                    1               2               1               0               0
    Route stats:
      Address family                       Received            Sent    Prefix limit    Idle timeout
      IPv4 unicast                                0               0               0               0
      IPv4 labeled-unicast                        0               0               0               0
      IPv6 unicast                                0               0               0               0
      IPv6 labeled-unicast                        0               0               0               0
      IPv4 VPN-unicast                            0               0               0               0
      IPv6 VPN-unicast                            0               0               0               0
      IPv4 multicast                              0               0               0               0
      IPv4 VPN-multicast                          0               0               0               0
      L2VPN VPLS                                  0               0               0               0
      L2VPN EVPN                                  0               0               0               0
      IPv4 Flowspec                               2               2               0               0
      IPv6 Flowspec                               1               1               0               0
```

Example 3: BGP Peer history for a specified peer

```
supervisor@rtbrick.net: op> show bgp peer history peer address 192:168::40
Instance: vrf1
  Peer Address                Source Address              Type              Last Reset
Reason
  192:168::40                 192:168:5::20               FSM Error         FSM Error,
Sub-Code: Unexpected message in OpenSent State
```

#### show bgp rib-in

This command displays the received routes.

**Syntax:**

**show bgp rib-in** <option> ...

| Option | Description |
|---|---|
| - | Without any option, the command displays information on the received BGP routing table on all instances in a summary table format. |
| <afi> | BGP routing table summary for the given address family (AFI), all sub-address families and all instances. Supported AFI values are 'ipv4' and 'ipv6'. |
| <afi> <safi> | BGP routing table summary for the given address family (AFI) and sub-address family (SAFI), and all instances. Supported SAFI values are 'labeled-unicast', 'unicast', 'vpn-multicast', 'vpn-unicast', ‘evpn-vpws’, ‘evpn’, ‘vpls-vpws’, ‘vpls’, and ‘flowspec’. |

| Option | Description |
|---|---|
| `<afi> <safi> detail` | Detailed list view of the BGP routing table for the given address family (AFI) and sub-address family (SAFI), and all instances. |
| `instance <instance-name> policy-accept` | Displays all BGP FlowSpec routes received in the specified address family (ipv4, ipv6, or l2vpn) that were accepted by the BGP import policy. |
| `instance <instance-name> policy-accept detail` | Detailed view of the BGP FlowSpec routes received in the specified address family (ipv4, ipv6, or l2vpn) that were accepted by the BGP import policy. |
| `instance <instance-name> policy-deny` | Displays all BGP FlowSpec routes received in the specified address family (ipv4, ipv6, or l2vpn) that were denied by the BGP import policy. |
| `instance <instance-name> policy-deny detail` | Detailed view of the BGP FlowSpec routes received in the specified address family (ipv4, ipv6, or l2vpn) that were denied by the BGP import policy. |
| `<afi> <safi> <prefix>` | BGP routing table entry for the given prefix and all instances. |
| `<afi> <safi> instance <instance-name>` | BGP routing table summary for the given AFI, SAFI, and instance. |
| `<afi> <safi> instance <instance-name> detail` | Detailed list view of BGP routing table for the given AFI, SAFI, and instance. |
| `<afi> <safi> instance <instance-name> <prefix>` | BGP routing table entry for the given prefix and instance. |
| `<afi> <safi> community <community-name>` | BGP community details for the given AFI, SAFI, and instance. |
| `<afi> <safi> error` | BGP route with error status for the given AFI, SAFI, and instance. |
| `<afi> <safi> peer <name> / peer address <ip>` | Peer name or address. |

Example 1: Summary view of the BGP rib-in for the ipv4 flowspec address family.

```
supervisor@rtbrick: op> show bgp rib-in ipv4 flowspec
Instance: default, AFI: ipv4, SAFI: flowspec
  Hostname: SN-STD-2-3966602, Peer IP: 192.0.2.102
  Source IP: 192.0.2.101, Total routes: 2
    Flowspec Hash                                Match                                        Action
AS Path
    521b87e1                                     dest-prefix : 198.51.100.91/32               rate-limit: 400.0 kBps
65549
                                                        src-prefix  : 198.51.100.92/32
                                                        ip-proto    : [==icmp ]
                                                        port        : [==49152 ]
                                                        dst-port    : [==49152 ]
                                                        src-port    : [==49152 ]
    ad2c54f5                                     dest-prefix : 198.51.100.91/32               rate-limit: 400.0 kBps
65549
                                                        src-prefix  : 198.51.100.92/32
                                                        ip-proto    : [==icmp ]
                                                        dst-port    : [==49152 ]
                                                        src-port    : [==49152 ]
```

Example 2: Summary view of the BGP rib-in for the ipv6 flowspec address family.

```
supervisor@rtbrick: op> show bgp rib-in ipv6 flowspec
Instance: default, AFI: ipv6, SAFI: flowspec
  Hostname: SN-STD-2-3966602, Peer IP: 192.0.2.102
  Source IP: 192.0.2.101, Total routes: 1
    Flowspec Hash                                Match                                      Action
AS Path
    7bfa7233                                     dest-prefix : 2001:db8:0:1::91/128         None
65549
                                                        src-prefix  : 2001:db8:0:1::92/128
                                                        ip-proto    : [==icmp ]
                                                        src-port    : [==49154 ]
```

Example 3: Summary view of the BGP rib-in for the IPv4 with the error flag.

```
supervisor@rtbrick>rtbrick.net: op> show bgp rib-in ipv4
Flags: & - Imported, ! - Error
Instance: default, AFI: ipv4, SAFI: unicast
  Hostname: Local, Peer IP: 0.0.0.0
  Source IP: 0.0.0.0, Total routes: 4
    Flags  Prefix           Next Hop  MED       Lpref       AS Path
           12.0.0.0/24      -         0         100         -
           12.1.0.0/24      -         0         100         -
           192.168.0.10/32  -         0         100         -
           192.168.0.11/32  -         0         100         -
  Hostname: P1, Peer IP: 12.0.0.2
  Source IP: 12.0.0.1, Total routes: 4
    Flags  Prefix           Next Hop  MED       Lpref       AS Path
           12.0.0.0/24      12.0.0.2  0         -           4200000002
           12.1.0.0/24      12.0.0.2  0         -           4200000002
           192.168.0.20/32  12.0.0.2  0         -           4200000002
           192.168.0.21/32  12.0.0.2  0         -           4200000002
```

#### show bgp rib-out

This command displays the send routes.

**Syntax:**

**show bgp rib-out** <option> ...

| Option | Description |
|---|---|
| - | Without any option, the command displays advertised BGP routes for all instances. |
| <afi> | BGP routing table summary for the given address family (AFI), all sub-address families and all instances. Supported AFI values are 'ipv4' and 'ipv6'. |

| Option | Description |
|---|---|
| `<afi> <safi>` | BGP routing table summary for the given address family (AFI) and sub-address family (SAFI), and all instances. Supported SAFI values are 'unicast', 'labeled-unicast', 'multicast', 'vpn-unicast', ‘evpn’, ‘vpls’, ‘vpls-vpws’, and ‘flowspec’. |
| `<afi> <safi> detail` | Detailed list view of the BGP routing table for the given address family (AFI) and sub-address family (SAFI), and all instances. |
| `<afi> <safi> <prefix>` | BGP routing table entry for the given prefix and all instances. |
| `<afi> <safi> instance <instance-name>` | BGP routing table summary for the given AFI, SAFI, and instance. |
| `<afi> <safi> instance <instance-name> detail` | Detailed list view of BGP routing table for the given AFI, SAFI, and instance. |
| `<afi> <safi> instance <instance-name> <prefix>` | BGP routing table entry for the given prefix and instance. |
| `<afi> <safi> peer <name> / peer address <ip>` | Peer name or address |

Example 1: Summary view of the IPv4 FlowSpec routes advertised to a peer

```
supervisor@rtbrick: op> show bgp rib-out ipv4 flowspec
Flags: N - RPKI Unknown, I - RPKI Invalid, V - RPKI Valid
Instance: default, AFI: ipv4, SAFI: flowspec
  Peer-group: FS, Sent routes: 2
   Flags  Flowspec Hash                Match                                      Action                      Origin
          521b87e1                     dest-prefix : 198.51.100.91/32             None
Incomplete
                                       src-prefix  : 198.51.100.92/32
                                       ip-proto    : [==icmp ]
                                       port        : [==49152 ]
                                       dst-port    : [==49152 ]
                                       src-port    : [==49152 ]
          ad2c54f5                     dest-prefix : 198.51.100.91/32             None
Incomplete
                                       src-prefix  : 198.51.100.92/32
                                       ip-proto    : [==icmp ]
                                       dst-port    : [==49152 ]
                                       src-port    : [==49152 ]
  Peer-group: PE2, Sent routes: 2
   Flags  Flowspec Hash                Match                                      Action                      Origin
          521b87e1                     dest-prefix : 198.51.100.91/32             None
Incomplete
                                       src-prefix  : 198.51.100.92/32
                                       ip-proto    : [==icmp ]
                                       port        : [==49152 ]
                                       dst-port    : [==49152 ]
                                       src-port    : [==49152 ]
          ad2c54f5                     dest-prefix : 198.51.100.91/32             None
Incomplete
                                       src-prefix  : 198.51.100.92/32
                                       ip-proto    : [==icmp ]
                                       dst-port    : [==49152 ]
                                       src-port    : [==49152 ]
```

Example 2: Summary view of the IPv6 FlowSpec routes advertised to a peer

```
supervisor@rtbrick: op> show bgp rib-out ipv6 flowspec
Flags: N - RPKI Unknown, I - RPKI Invalid, V - RPKI Valid
Instance: default, AFI: ipv6, SAFI: flowspec
  Peer-group: FS, Sent routes: 1
   Flags  Flowspec Hash                Match                                      Action                      Origin
          7bfa7233                     dest-prefix : 2001:db8:0:1::91/128         None
Incomplete
                                       src-prefix  : 2001:db8:0:1::92/128
                                       ip-proto    : [==icmp ]
                                       src-port    : [==49154 ]
  Peer-group: PE2, Sent routes: 1
   Flags  Flowspec Hash                Match                                      Action                      Origin
          7bfa7233                     dest-prefix : 2001:db8:0:1::91/128         None
Incomplete
                                       src-prefix  : 2001:db8:0:1::92/128
                                       ip-proto    : [==icmp ]
                                       src-port    : [==49154 ]
```

#### show bgp fib

The 'show bgp fib' commands display the BGP forwarding table. In contrast to the 'show bgp rib' commands, the output of the 'show bgp fib' commands includes only the selected routes. The BGP route selection occurs between the RIB and the FIB.

**Syntax:**

**show bgp fib** <option> ...

| Option | Description |
|---|---|
| - | Without any option, the commands display the BGP forwarding table for all address families and all instances in a summary table format. |
| afi> | BGP forwarding table summary for the given address family (AFI), all sub-address families and all instances. Supported AFI values are 'ipv4' and 'ipv6'. |
| <afi> <safi> | BGP forwarding table summary for the given address family (AFI) and sub-address family (SAFI), and all instances. Supported SAFI values are 'unicast', 'labeled-unicast', 'vpn-multicast', 'vpn-unicast', ‘evpn-vpws’, ‘vpls’, ‘vpls-vpws’ and ‘flowspec’. |
| <afi> <safi> detail | Detailed list view of the BGP forwarding table for the given address family (AFI) and sub-address family (SAFI), and all instances. |
| <afi> <safi> <prefix> | BGP forwarding table entry for the given prefix and all instances. |
| <afi> <safi> instance <instance-name> | BGP forwarding table summary for the given AFI, SAFI, and instance. |
| <afi> <safi> instance <instance-name> detail | Detailed list view of BGP forwarding table for the given AFI, SAFI, and instance. |

| Option | Description |
| --- | --- |
| <afi> <safi> instance <instance-name> <prefix> | BGP forwarding table entry for the given prefix and instance. |

Example 1: Summary view of the BGP FIB for IPv4 flowspec address family

```
supervisor@rtbrick: op> show bgp fib ipv4 flowspec
Instance: default, AFI: ipv4, SAFI: flowspec, Total routes: 2
  Flowspec Hash            Match                                      Action                    Priority   Status
  521b87e1                 dest-prefix : 198.51.100.91/32             rate-limit: 400.0 kBps    1501       Ignored
                           src-prefix  : 198.51.100.92/32
                           ip-proto    : [==icmp ]
                           port        : [==49152 ]
                           dst-port    : [==49152 ]
                           src-port    : [==49152 ]
  ad2c54f5                 dest-prefix : 198.51.100.91/32             rate-limit: 400.0 kBps    1502       Valid
                           src-prefix  : 198.51.100.92/32
                           ip-proto    : [==icmp ]
                           dst-port    : [==49152 ]
                           src-port    : [==49152 ]
```

Example 2: Summary view of the BGP FIB for a specific IPv4 flowspec address family

```
supervisor@rtbrick: op> show bgp fib ipv4 flowspec ad2c54f5
Instance: default, AFI: ipv4, SAFI: flowspec, Total routes: 1
  Flowspec hash: ad2c54f5
    Match:
      dest-prefix:  198.51.100.91/32
      src-prefix:  198.51.100.92/32
      ip-proto:  [==icmp ]
      dst-port:  [==49152 ]
      src-port:  [==49152 ]
    Action:
      rate-limit: 400.0 kBps
    Extended community:
      ['traffic-rate-bytes:0:400000.000000']
    Priority: 1502
    Status: Valid
    Number of ACL added: 1
```

Example 3: Summary view of the BGP FIB for IPv6 flowspec address family

```
supervisor@rtbrick: op> show bgp fib ipv6 flowspec
Instance: default, AFI: ipv6, SAFI: flowspec, Total routes: 1
  Flowspec Hash            Match                                      Action                    Priority   Status
  7bfa7233                 dest-prefix : 2001:db8:0:1::91/128         None                      1501       Valid
                           src-prefix  : 2001:db8:0:1::92/128
                           ip-proto    : [==icmp ]
                           src-port    : [==49154 ]
```

Example 4: Summary view of the BGP FIB for a specific IPv6 flowspec address family

```
supervisor@rtbrick: op> show bgp fib ipv6 flowspec 7bfa7233
Instance: default, AFI: ipv6, SAFI: flowspec, Total routes: 1
  Flowspec hash: 7bfa7233
    Match:
      dest-prefix:  2001:db8:0:1::91/128
      src-prefix:  2001:db8:0:1::92/128
      ip-proto:  [==icmp ]
      src-port:  [==49154 ]
    Action:
      None
    Extended community:
      ['None']
    Priority: 1501
    Status: Valid
    Number of ACL added: 1
```

#### Show Command Filter Options for RIB-in, RIB-out, and FIB

### Syntax

**show bgp fib|rib-in|rib-out** <afi> **flowspec filter** <options>

| Option | Description |
|---|---|
| `<afi> flowspec filter <options>` | Filter BGP Flowspec entries across the RIB based on the option specified. |
| `destination-port` | Filters based on destination port number. |
| `destination-prefix` | Filters based on the destination IP prefix. |
| `instance-name` | Filters entries based on the BGP instance. Useful to verify the Flowspec entries specific to one instance in a multi-instance BGP configurations. |
| `ip-proto` | Filters based on the protocol. |
| `port` | Filters Flowspec entries based on the protocol port number. |
| `source-port` | Filters entries based on the source port. Useful to identify different applications or services originating from specific ports. |
| `source-prefix` | Filters entries based on the source IP prefix. |

Example 1: Filter BGP RIB-in Flowspec entries based on the port

```
supervisor@rtbrick: op> show bgp rib-in ipv4 flowspec filter port 49152
Instance: default, AFI: ipv4, SAFI: flowspec
  Hostname: SN-STD-2-3966602, Peer IP: 192.0.2.102
  Source IP: 192.0.2.101, Total routes: 1
    Flowspec Hash                              Match                                      Action
AS Path
    521b87e1                                   dest-prefix : 198.51.100.91/32             rate-limit: 400.0 kBps
```

```
65549
                                                        src-prefix  : 198.51.100.92/32
                                                        ip-proto    : [==icmp ]
                                                        port        : [==49152 ]
                                                        dst-port    : [==49152 ]
                                                        src-port    : [==49152 ]
```

Example 2: Filter BGP RIB-in Flowspec entries based on destination prefix

```
supervisor@rtbrick: op> show bgp rib-in ipv4 flowspec filter destination-prefix 198.51.100.91/32
Instance: default, AFI: ipv4, SAFI: flowspec
  Hostname: SN-STD-2-3966602, Peer IP: 192.0.2.102
  Source IP: 192.0.2.101, Total routes: 2
    Flowspec Hash                                Match                                       Action
AS Path
    521b87e1                                     dest-prefix : 198.51.100.91/32              rate-limit: 400.0 kBps
65549
                                                        src-prefix  : 198.51.100.92/32
                                                        ip-proto    : [==icmp ]
                                                        port        : [==49152 ]
                                                        dst-port    : [==49152 ]
                                                        src-port    : [==49152 ]
    ad2c54f5                                     dest-prefix : 198.51.100.91/32              rate-limit: 400.0 kBps
65549
                                                        src-prefix  : 198.51.100.92/32
                                                        ip-proto    : [==icmp ]
                                                        dst-port    : [==49152 ]
                                                        src-port    : [==49152 ]
```

Example 3: Filter BGP RIB-in Flowspec entries based on source prefix

```
supervisor@rtbrick: op> show bgp rib-in ipv4 flowspec filter source-prefix 198.51.100.92/32
Instance: default, AFI: ipv4, SAFI: flowspec
  Hostname: SN-STD-2-3966602, Peer IP: 192.0.2.102
  Source IP: 192.0.2.101, Total routes: 2
    Flowspec Hash                                Match                                       Action
AS Path
    521b87e1                                     dest-prefix : 198.51.100.91/32              rate-limit: 400.0 kBps
65549
                                                        src-prefix  : 198.51.100.92/32
                                                        ip-proto    : [==icmp ]
                                                        port        : [==49152 ]
                                                        dst-port    : [==49152 ]
                                                        src-port    : [==49152 ]
    ad2c54f5                                     dest-prefix : 198.51.100.91/32              rate-limit: 400.0 kBps
65549
                                                        src-prefix  : 198.51.100.92/32
                                                        ip-proto    : [==icmp ]
                                                        dst-port    : [==49152 ]
                                                        src-port    : [==49152 ]
```

Example 4: Filter BGP RIB-in Flowspec entries based on destination port

```
supervisor@rtbrick: op> show bgp rib-in ipv4 flowspec filter destination-port 49152
Instance: default, AFI: ipv4, SAFI: flowspec
  Hostname: SN-STD-2-3966602, Peer IP: 192.0.2.102
  Source IP: 192.0.2.101, Total routes: 2
    Flowspec Hash                                Match                                       Action
AS Path
```

```
    521b87e1                                dest-prefix : 198.51.100.91/32              rate-limit: 400.0 kBps
65549
                                                   src-prefix  : 198.51.100.92/32
                                                   ip-proto    : [==icmp ]
                                                   port        : [==49152 ]
                                                   dst-port    : [==49152 ]
                                                   src-port    : [==49152 ]
    ad2c54f5                                dest-prefix : 198.51.100.91/32              rate-limit: 400.0 kBps
65549
                                                   src-prefix  : 198.51.100.92/32
                                                   ip-proto    : [==icmp ]
                                                   dst-port    : [==49152 ]
                                                   src-port    : [==49152 ]
```

Example 5: Filter BGP RIB-in Flowspec entries based on source port

```
supervisor@rtbrick: op> show bgp rib-in ipv4 flowspec filter source-port 49152
Instance: default, AFI: ipv4, SAFI: flowspec
  Hostname: SN-STD-2-3966602, Peer IP: 192.0.2.102
  Source IP: 192.0.2.101, Total routes: 2
    Flowspec Hash                           Match                                       Action
AS Path
    521b87e1                                dest-prefix : 198.51.100.91/32              rate-limit: 400.0 kBps
65549
                                                   src-prefix  : 198.51.100.92/32
                                                   ip-proto    : [==icmp ]
                                                   port        : [==49152 ]
                                                   dst-port    : [==49152 ]
                                                   src-port    : [==49152 ]
    ad2c54f5                                dest-prefix : 198.51.100.91/32              rate-limit: 400.0 kBps
65549
                                                   src-prefix  : 198.51.100.92/32
                                                   ip-proto    : [==icmp ]
                                                   dst-port    : [==49152 ]
                                                   src-port    : [==49152 ]
```

Example 6: Filter BGP RIB-in Flowspec entries based on IP protocol

```
supervisor@rtbrick: op> show bgp rib-in ipv4 flowspec filter ip-proto icmp
Instance: default, AFI: ipv4, SAFI: flowspec
  Hostname: SN-STD-2-3966602, Peer IP: 192.0.2.102
  Source IP: 192.0.2.101, Total routes: 2
    Flowspec Hash                           Match                                       Action
AS Path
    521b87e1                                dest-prefix : 198.51.100.91/32              rate-limit: 400.0 kBps
65549
                                                   src-prefix  : 198.51.100.92/32
                                                   ip-proto    : [==icmp ]
                                                   port        : [==49152 ]
                                                   dst-port    : [==49152 ]
                                                   src-port    : [==49152 ]
    ad2c54f5                                dest-prefix : 198.51.100.91/32              rate-limit: 400.0 kBps
65549
                                                   src-prefix  : 198.51.100.92/32
                                                   ip-proto    : [==icmp ]
                                                   dst-port    : [==49152 ]
                                                   src-port    : [==49152 ]
```

#### Validating FlowSpec ACLs

In forwarding, the FlowSpec rules can be validated using the "show acl rule <...>" command as shown in the example below:

Example: Validate BGP FlowSpec ACL rule

```
supervisor@rtbrick: op> show acl rule bgp-flowspec-ad2c54f59cf8944514bcfb21a0e2beec40a45efaea900442
Rule: bgp-flowspec-ad2c54f59cf8944514bcfb21a0e2beec40a45efaea900442
  ACL type: ext_l3v4
  Ordinal: 0             Priority: 1502
    Match:
      Direction: external
      Instance: default
      Destination IPv4 prefix: 198.51.100.91/32
      Source IPv4 prefix: 198.51.100.92/32
      Destination L4 port: 49152
      Source L4 port: 49152
      IP protocol: icmp
    Action:
      Policer profile name: bgp-flowspec-ad2c54f59cf8944514bcfb21a0e2beec40a45efaea900442
    Result:
      ACL Handle: 5
      External:
        Value: 80224256
          Is Trap ID: No
          Trap ID: 0
          Trap Strength: 0
          Is QoS: Yes
          Class: 0
          Policer ID: 612
```

### BGP FlowSpec Clear Commands

Clear commands allow to reset operational states.

#### BGP Peer

This commands resets BGP peerings.

Syntax:

**clear bgp peer** <option> ...

| Option | Description |
|---|---|
| `all` | Clears all the BGP peers. |
| `all soft-in <afi> <safi>` | Sends route refresh to all neighbors. |
| `all soft-out <afi> <safi>` | Re-advertises all the routes previously sent to the peer. |
| `all stats` | Clears the statistics of all the BGP peers. |

| Option | Description |
|---|---|
| `instance <instance> <peer-ip>` | Clears the peer for the given instance and peer IP address. |
| `instance <instance> <peer-ip> source <src-ip>` | Clears a specific peer for the given peer IP address and source IP address in the specified instance. |
| `instance <instance> all` | Clears all peers in the given instance. |
| `instance <instance> <peer-ip> source <src-ip> soft-in <afi> <safi>` | Sends route refresh to specific peer for the given instance, peer-ip, source-ip and address-family. |
| `instance <instance> <peer-ip> soft-in <afi> <safi>` | Sends route refresh to peer for the given instance, peer-ip and address-family. |
| `instance <instance> all soft-in <afi> <safi>` | Sends route refresh to all peers for the given instance and address family. |
| `instance <instance> <peer-ip> source <src-ip> soft-out <afi> <safi>` | Re-advertises all the routes previously sent to the specific peer for the given instance, peer-ip, source-ip and address-family. |
| `instance <instance> <peer-ip> soft-out <afi> <safi>` | Sends route refresh to peer for a given instance, peer-ip and address-family. |
| `instance <instance> all soft-out <afi> <safi>` | Sends route update to all peers for given instance and address family. |
| `instance <instance> <peer-ip> source <src-ip> stats` | Clears the statistics of a specific peer for a given instance, peer-ip and source-IP. |
| `instance <instance> <peer-ip> stats` | Clears the statistics of the peer for a given instance and peer-IP. |
| `instance <instance> all stats` | Clears the statistics of all peers for a given instance. |

Example: The example below shows how to clear all the BGP peers.

```
supervisor@rtbrick: op> clear bgp peer all
```

Example: Route Refresh in IPv4/IPv6 for BGP FlowSpec

```
supervisor@rtbrick: op> clear bgp peer all soft-out ipv4 flowspec
supervisor@rtbrick: op> clear bgp peer all soft-in ipv6 flowspec
```

# 2.5. BGP RPKI-RTR Route Validation

## 2.5.1. BGP RPKI-RTR Prefix Validation Overview

BGP is the core of the Internet, serving as the backbone protocol that enables exchange of traffic between networks. It plays a crucial role in connecting autonomous systems and ensuring reliable data transmission across the Internet. However, BGP has some inherent security limitations due to its design. It cannot verify whether incoming BGP advertisements come from an authorized

autonomous system (AS). This limitation leaves BGP open to various attacks such as route hijacking.

**Challenges Due to False Prefix Announcements**

**Route Hijacking**

An attacker can falsely advertise ownership of an IP prefix to redirect or intercept traffic. Also, an autonomous system can incorrectly advertise routes it learned from one provider to other providers. This can be an accidental route leak that directs routes to unintended destinations.

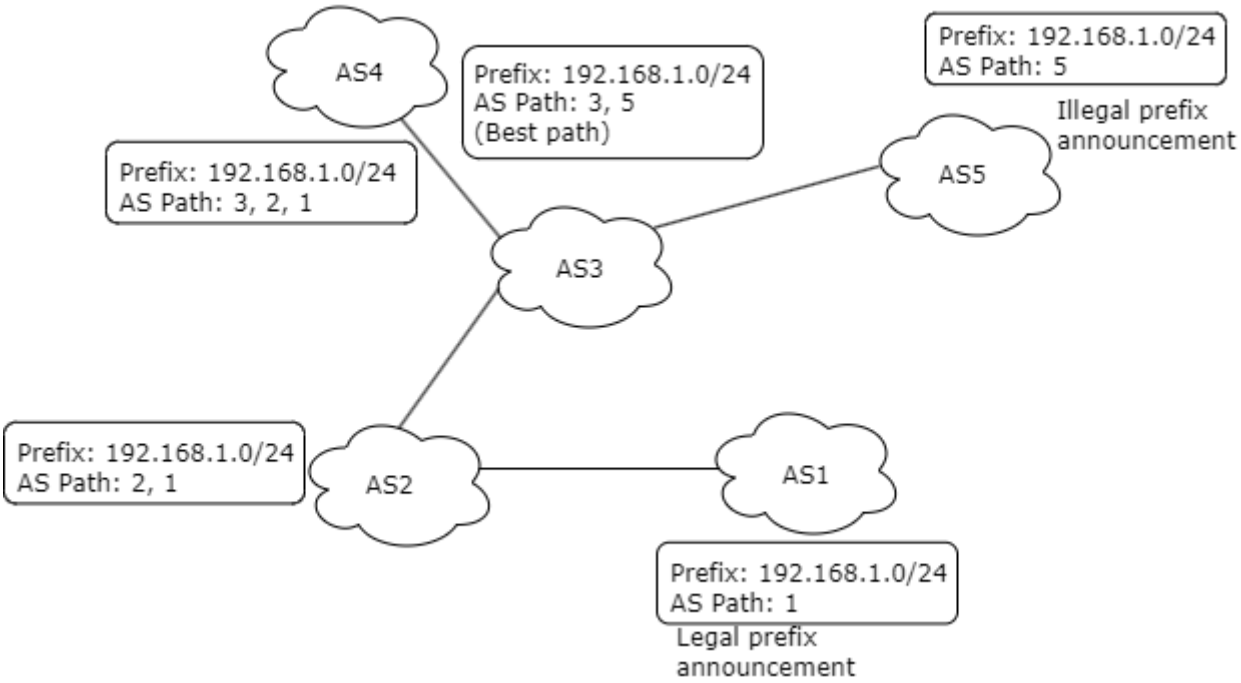



*Figure 10. Route Hijacking*

The preceding image is an example of a prefix hijacking scenario. ‘AS1’ announces 192.168.1.0/24, while an attacker announces the same prefix from ‘AS5’. ‘AS1’ is the only authorized to originate and announce this prefix on the Internet. ‘AS5’, without authorization, sends a BGP announcement for the same prefix, 192.168.1.0/24. As a result, other ASes on the Internet receive two conflicting announcements for this prefix, one from 'AS1' and another from 'AS5'. AS4 prefers AS5’s announcement due to the shortest AS path. So, AS5’s unauthorized announcement is selected as the best path; traffic intended for 192.168.1.0/24 will be rerouted to 'AS5' instead of 'AS1'.

Route Hijacking occurs in a number of ways:

**Prefix Hijacking:** An AS can advertise IP prefixes that it does not legitimately own. Attacker can intercept traffic for those prefixes.

**AS Path Hijacking:** In this scenario, an AS modifies the AS path of a route advertisement to attract traffic that would normally pass through another AS. This can lead to malicious traffic interception or disruption.

**Misconfiguration:** A simple configuration error can cause large-scale traffic redirection that affects the internet connectivity.

RBFS BGP implements RPKI-RTR based prefix validation mechanism that verifies the authenticity of

route announcements before they are accepted and propagated.

### Understanding BGP RPKI-RTR Validation Implementation

Resource Public Key Infrastructure (RPKI) is a security framework to protect BGP routes from specific types of attacks like route hijacking and misconfigurations. It provides a way to verify the origin AS of BGP routes, ensuring that only authorized ASs are allowed to advertise specific IP prefixes. It achieves this by cryptographically associating IP address allocations (both IPv4 and IPv6 prefixes) with their rightful owner ISPs or enterprises.

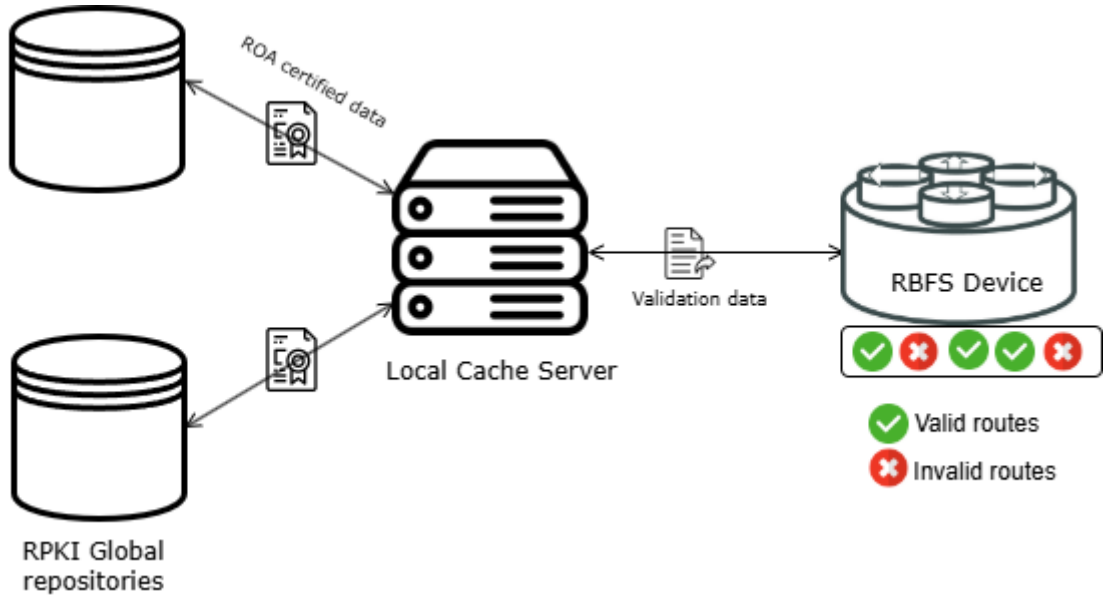



*Figure 11. Implementation of BGP RPKI-RTR Validation*

#### Key Components of the Validation Mechanism

### Global RPKI Repository

Global RPKI servers are maintained by Regional Internet Registries and other trusted entities. These servers contain cryptographic certificates and Route Origin Authorizations (ROAs) for IP prefixes and ASNs. This is the authoritative source where the reliable information about IP address ownership.

### Local Cache Server

ISPs deploy local cache servers, which are configured to regularly synchronize with global RPKI repositories, mirroring the latest ROAs and certified data. The local cache servers serve as intermediaries between the global repository and RBFS devices.

### RBFS Device

Once configured, RBFS device establishes secure communication channel with the local cache server for fetching the data using the RTR protocol. Once fetched, the RPKI data is stored locally within the RBFS system. Data are stored in a data table with entries that correspond to valid route origin authorizations. It allows quick lookups for route validation when the device receives a new prefix announcement from a BGP peer.

#### How RPKI Validation Works

- `RPKI Data Repos`: Network operators create ROAs which cryptographically bind an IP prefix with an authorized ASN. ROAs are stored in global RPKI repositories which are trusted sources are accessible globally. Local caches servers access the data and stores the verified data.
- `RPKI Data`: RBFS retrieves of data from cache servers. After establishing a validation session with a cache server, RBFS periodically sends reset queries at refresh intervals. These queries request the complete set of records from the cache. When RBFS receives an end-of-data (EOD) PDU, it performs garbage collection to remove outdated records. RBFS maintains the verified data from the local cache servers and stores the data within the system.
- `Route Announcement`: When a new prefix is advertised from a peer, RBFS receives the advertisement.
- `Validation`: RBFS looks up and verifies the ROA to see if the origin AS in the BGP route matches the authorized AS in the ROA using data stored in its RPKI data table. The result of this verification can be 'Valid', 'Invalid', or 'Unknown'.
- `Decision`: Based on the configured validation filter ('strict', 'loose', ,'mark-only' or 'disable'), the RBFS decides whether to accept or reject the route.

Once the route is validated, RBFS follows decision process as follows:

- `Valid Routes`: Valid routes are installed into the routing table and used for traffic forwarding.
- `Invalid Routes`: Invalid routes are discarded if you have configured the filter option as 'loose' or 'strict'. If the filter option is configured with 'mark-only', the routes are accepted and will have to processed in routing policy.
- `Unknown`: Routes that are not found in the RPKI data will be accepted if you have configured the filter option as `loose`. Otherwise, this will be discarded.
- `Init`: Routes learned from BGP sessions that have no validation configured or where the RTR session is down, are marked as `init`.

This entire process ensures that only verified and authorized routes are propagated within the routing tables.

> IP prefixes may change any time. Once configured, an RBFS device continuously validates prefixes in real-time when new BGP route announcements arrive. If the RPKI data changes (for example, a new ROA is added or updated), the RBFS device needs to revalidate routes which were accepted or rejected previously. This ensures that only routes verified through RPKI are accepted and malicious route announcements are rejected.

#### Shared Validation Databases

RBFS allows BGP prefixes in one instance to use the validation database from another VRF instance. It allows for sharing the validation database between routing instances within the system. When a prefix is received in a BGP update, the receiving BGP instance checks its validity by querying the shared database.

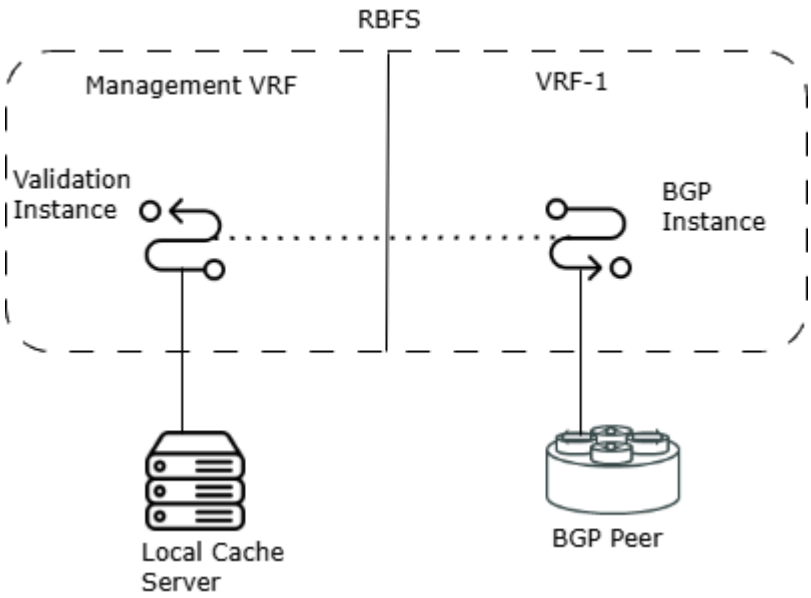



*Figure 12. Shared Validation Databases*

**Timers**

With timers, RBFS ensures that the router maintains an up-to-date RPKI validation database. Timers are used to periodically refresh, retry, and expire validation data based on these intervals. RBFS uses timers to maintain synchronization between local cache and RBFS device.

**Refresh Interval**: Specifies how frequently the router should query the Local Cache to refresh its RPKI validation database. Router polls the cache with Reset Query PDU at Refresh intervals.

**Retry Interval**: Specifies how long the router should wait before retrying after a failed Reset Query PDU. Router sends Reset Query PDU upon the failed reset query at retry interval.

**Expire Interval**: Defines how long the router can continue using the current version of the validation database if communication with the Local Cache fails. Router uses the current version of validation database until expire interval.

RBFS clears the validation session and the database after three consecutive unsuccessful reset query attempts at the refresh interval.

**Supported Platforms**

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

## 2.5.2. BGP Prefix Validation Using RPKI-RTR Configuration

**Configuration Hierarchy**

The diagram illustrates the BGP prefix validation mechanism configuration hierarchy.

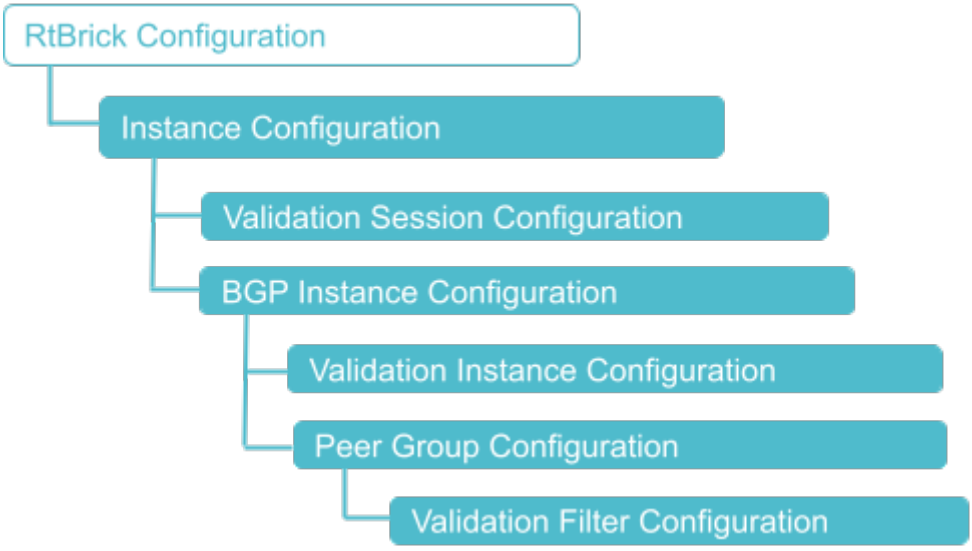



*Figure 13. BGP RPKI Configuration Hierarchy*

### Configuration Syntax and Commands

The following sections describe the BGP RPKI prefix validation configuration syntax and commands.

#### Configure Validation Session with RPKI Local Cache

The following configuration commands are used to define validation session on an RBFS device. It specifies different session parameters such as cache IP address, update source, port, and preference.

In scenarios where multiple caches are available, the system employs a deterministic method for selecting the preferred cache. The cache preference method provides a way to determine which cache should be used based on configurable priorities.

**Cache Preference:**

1. When a preference is configured, the cache with the numerically lower preference value is selected.
2. If no preference is configured, or two caches have the same preference value, the selection follows these rules:
    - Among IPv4 caches, the cache with the higher IP address is preferred.
    - Among IPv6 caches, the cache with the higher IP address is preferred.
    - Between IPv4 and IPv6 caches, the IPv4 cache takes precedence over IPv6.

**Syntax**

```
set instance <instance-name> protocol validation session [IPv4|IPv6] <cache-address>
<update-source>
```

| Attribute | Description |
|---|---|
| `cache-address` | Cache server IP address. |
| `update-source` | Specify the source IP address. This address is used to establish a TCP connection with RPKI cache to fetch RPKI data. |
| `port` | Port number used for the RPKI local cache. Range 1 - 65535. Default port is 3323. If you do not configure a port, the default port number is used. |
| `preference` | Enables RBFS to select the validated route record from the preferred cache. Range 1 - 255. A numerically lower preference value indicates the higher preference. |

Example: Validation session with the RPKI local cache

The following configuration sets up RPKI validation for both IPv4 and IPv6 routes in BGP. IPv4 session connects the router to an RPKI cache server with the RTR protocol, using the specified cache server IP address 192.168.5.50. The update-source, that is source IP address, is specified to 192.168.5.20 for the IPv4 session. The port number that is used for RTR communication is specified to '3345' for the session.

For the IPv6 session, the cache server IP address is specified to 192:168:5::50 and the source IP is specified 192:168:5::20. The port number is specified 3346 that is used for RTR communication.

The preference value for the IPv4 session is set to 100, while the preference value for the IPv6 session is set to 10. The preference value determines which session is prioritized when both sessions are available. A lower numerical preference value indicates a higher preference. In this scenario, the IPv6 session has a higher preference because it has a lower value, meaning that the IPv6 session will take priority over the IPv4 session.

```
set instance vrf-blue protocol validation session ipv4 192.168.5.50 192.168.5.20
set instance vrf-blue protocol validation session ipv6 192:168:5::50 192:168:5::20
port 3345
set instance vrf-blue protocol validation session ipv4 192.168.5.50 192.168.5.20
preference 100
set instance vrf-blue protocol validation session ipv6 192:168:5::50 192:168:5::20
set instance vrf-blue protocol validation session ipv6 192:168:5::50 192:168:5::20
port 3346
set instance vrf-blue protocol validation session ipv6 192:168:5::50 192:168:5::20
preference 10
```

The following example shows the validation session with RPKI local cache.

```
supervisor@rtbrick.net: cfg> show config instance vrf-blue protocol validation
{
  "rtbrick-config:validation": {
    "session": {
      "ipv4": [
```

```
        {
          "cache-address": "192.168.5.50",
          "update-source": "192.168.5.20",
          "port": 3345,
          "preference": 100
        }
      ],
      "ipv6": [
        {
          "cache-address": "192:168:5::50",
          "update-source": "192:168:5::20",
          "port": 3346,
          "preference": 10
        }
      ]
    }
  }
}
```

To access the RESTCONF API that corresponds to this CLI, click here.

**Configure BGP Validation Instance**

**Syntax**

**set instance** <instance-name> **protocol bgp validation-instance** <validation-instance>

| Attribute | Description |
|---|---|
| `<validation-instance>` | Validation instance name. |

Example: Validation Instance configuration

The following configuration sets up 'vrf-blue' as a validation instance.

```
set instance default protocol bgp validation-instance vrf-blue
```

The following example shows validation instance as vrf-blue.

```
supervisor@rtbrick.net: cfg> show config instance default protocol bgp validation-
instance
{
  "rtbrick-config:validation-instance": "vrf-blue"
}
```

#### Configure BGP Validation Filter

Validation filter can be configured under a specific BGP peer group.

**Syntax**

**set instance** <instance-name> **protocol bgp peer-group** <pg-name> **validation-filter** [strict|loose|mark-only|disable]

| Attribute | Description |
|---|---|
| `strict` | BGP accepts only the valid routes. |
| `loose` | BGP accepts both valid routes and unknown routes. But it rejects routes with invalid RPKI data. |
| `mark-only` | BGP will accept valid, invalid and unknown routes. All RPKI validation state processing must be done in routing policy. |
| `disable` | No route validation is performed. Routes are installed based on the BGP route selection process. |

Example: Validation filter configuration

The following configuration defines the validation filter. The validation filter is defined as 'loose' that means BGP can accept both valid routes and unknown routes, but it rejects routes with invalid RPKI data.

```
set instance default protocol bgp peer-group P2 validation-filter loose
```

The following example shows the validation filter as "loose.""

```
supervisor@rtbrick.net: cfg> show config instance default protocol bgp peer-group P2
validation-filter
{
  "rtbrick-config:validation-filter": "loose"
}
```

The following example shows the configuration of BGP validation filter "mark-only."

```
set instance default protocol bgp peer-group pg1 validation-filter mark-only
```

```
supervisor@rtbrick>LEAF01: cfg> show config instance default protocol bgp peer-group pg1 validation-filter
{
  "rtbrick-config:validation-filter": {
    "mark-only": "true"
  }
```

```
}
```

#### Ignoring RPKI Validation State When Selecting BGP Best Route

RBFS can be configured to ignore the RPKI Validation State while selecting the BGP best route. It is recommended to exercise caution when doing this, as this can cause unexpected results such as black holes and routing loops.

> Both "validation-filter mark-only" and "ignore-validation-state true" must be configured to enable proper processing in policy.

**Syntax:**

**set instance** <instance-name> **protocol bgp bestroute-selection ignore-validation-state** < **false** | **true** >

The following example shows the configuration for ignoring the RPKI validation state.

```
set instance default protocol bgp bestroute-selection ignore-validation-state true
```

```
supervisor@rtbrick>LEAF01: cfg> show config instance default protocol bgp bestroute-selection
{
  "rtbrick-config:bestroute-selection": {
    "ignore-validation-state": "true"

  }
}
```

### 2.5.3. BGP Route Validation with RPKI-RTR Operational Commands

#### BGP RPKI-RTR Show Commands

The BGP RPKI-RTR show commands provide detailed information about the BGP RPKI-RTR route validation operations.

#### Validation Session

The `show validation session` command and options provide information about the TCP session information between RBFS and the local caches.

**Syntax:**

**show validation session** <option>

| Option | Description |
|---|---|
| `-` | Without any option, it displays information about all the validation session. |
| `cache` | Displays validation session information filtered by a specific cache address. |
| `detail` | Provides detailed information about each validation session. |
| `instance` | Displays validation session information for a specific instance. |
| `source` | Displays validation session information filtered by a specific source IP address. |

Example: Validation session

```
supervisor@rtbrick: op> show validation session
Instance: default
  Cache                       Source                    State         Up/Down Time      IPv4 PDU Rcvd    IPv6 PDU Rcvd
  192.0.2.102                 192.0.2.101               Established   0d:00h:04m:06s           658188           188630
  2001:db8:101:1::2           2001:db8:101:1::1         Established   0d:00h:04m:06s           658188           188630
```

API To access the Operational State API that corresponds to this CLI, click include::common:partial$operational_api_links.adoc[tag=show_validation].

#### Validation Cache-in

The `show validation cache-in` command provides cache information.

**Syntax:**

**show validation cache-in** <option>

The show command and options provide detailed information about the received validation route records from the local caches.

| Option | Description |
|---|---|
| `-` | Without any option, it displays information about the prefix validation data for all caches and source IPs. |
| `afi` | Filters the cache-in validation data based on the address family, such as IPv4 or IPv6. |
| `cache` | Displays validation information related to a specific cache address. |
| `instance` | Filters the validation data for a specific instance. |
| `prefix` | Displays validation information for a specific prefix. |

API To access the Operational State API that corresponds to this CLI, click include::common:partial$operational_api_links.adoc[tag=show_validation].

Example: Information for the specific cache.

```
supervisor@rtbrick: op> show validation cache-in instance default cache 192.0.2.102 | head -n 15
Instance: default, AFI: ipv4
  Cache IP: 192.0.2.102, Source IP: 192.0.2.101, Total Prefixes: 658188
    Prefix                                      As Num    Max Len
    80.128.0.0/11                                    0         11
    96.160.0.0/11                                    0         11
    166.240.0.0/12                                   0         12
    174.8.0.0/13                                     0         13
    174.80.0.0/13                                    0         13
    69.96.0.0/13                                     0         13
    166.152.0.0/13                                   0         13
    166.160.0.0/13                                   0         13
    31.64.0.0/14                                     0         14
    97.68.0.0/14                                     0         14
    31.72.0.0/14                                     0         14
    23.84.0.0/14                                     0         14
```

#### Validation Database

The validation database contains preferred validation route records based the cache preference. The `show validation database` provides detailed information about the validation database, which is used for prefix validation.

**Syntax:**

**show validation database** <option>

| Option | Description |
|---|---|
| `-` | Without any option, it displays the information about all validation databases. |
| `afi` | Filters the validation database based on the address family, such as IPv4 or IPv6. |
| `as-num` | Displays validation information for a specific Autonomous System (AS). |
| `cache` | Displays database entries associated with a specific cache address. |
| `instance` | Displays database information for a specific instance. |
| `prefix` | Filters the database to show validation details for a specific prefix. |
| `summary` | Provides a summary of the validation database. |

API To access the Operational State API that corresponds to this CLI, click include::common:partial$operational_api_links.adoc[tag=show_validation].

Example 1: Validation database information

```
supervisor@rtbrick: op> show validation database | head -n 15
Instance: default, AFI: ipv4
  Prefix                                    Cache                As Num    Max Len
  1.0.0.0/24                                2001:db8:101:1::2     13335         24
  3.0.0.0/10                                2001:db8:101:1::2     16509         10
  3.0.0.0/15                                2001:db8:101:1::2     16509         15
  5.0.0.0/19                                2001:db8:101:1::2     29256         19
  5.0.0.0/21                                2001:db8:101:1::2     29256         21
  12.0.0.0/22                               2001:db8:101:1::2      6269         22
  20.0.0.0/11                               2001:db8:101:1::2      8075         11
  23.0.0.0/12                               2001:db8:101:1::2     20940         12
  23.0.0.0/24                               2001:db8:101:1::2     20940         24
  24.0.0.0/12                               2001:db8:101:1::2      7922         12
  24.0.0.0/16                               2001:db8:101:1::2     33659         16
  27.0.0.0/22                               2001:db8:101:1::2     16509         24
  31.0.0.0/16                               2001:db8:101:1::2      8374         16
```

Example 2: Validation database information for a specific AS number.

```
supervisor@rtbrick: op> show validation database as-num 13335 | head -n 15
Instance: default, AFI: ipv4
  Prefix                                    Cache                As Num    Max Len
  1.0.0.0/24                                2001:db8:101:1::2     13335         24
  104.16.0.0/20                             2001:db8:101:1::2     13335         20
  104.17.0.0/20                             2001:db8:101:1::2     13335         20
  104.18.0.0/20                             2001:db8:101:1::2     13335         20
  117.18.0.0/20                             2001:db8:101:1::2     13335         24
  104.19.0.0/20                             2001:db8:101:1::2     13335         20
  104.20.0.0/20                             2001:db8:101:1::2     13335         20
  143.20.0.0/24                             2001:db8:101:1::2     13335         24
  104.21.0.0/19                             2001:db8:101:1::2     13335         19
  104.21.0.0/20                             2001:db8:101:1::2     13335         20
  104.22.0.0/20                             2001:db8:101:1::2     13335         20
  104.22.0.0/24                             2001:db8:101:1::2     13335         24
  104.24.0.0/20                             2001:db8:101:1::2     13335         20
```

Example 3: Validation database summary

```
supervisor@rtbrick: op> show validation database summary | head -n 15
Instance: default
  IPv4 Prefix: 575747
  IPv6 Prefix: 160567
```

### Clear Commands

#### Clear Validation Session

The `clear validation session` command and options are used to remove or reset data related to validation session.

**Syntax:**

**clear validation session** <option>

| Option | Description |
|---|---|
| `all` | Resets all existing validation sessions (without clearing the database). |
| `all soft` | Refreshes the only the validation database by sending a reset query to the cache. |
| `instance` | Resets validation sessions per instance. |

Example: Clearing validation session options

```
clear validation session all
clear validation session all soft
clear validation session instance vrf-blue
clear validation session instance vrf-blue 192.168.5.50 source 192.168.5.20 soft
```

API To access the Operational State API that corresponds to this CLI, click include::common:partial$operational_api_links.adoc[tag=clear_validation_session].

#### Clear Validation Database

The `clear validation database` command is used to remove entries in the validation database.

**Syntax:**

**clear validation database** <option>

| Option | Description |
|---|---|
| `all` | Clears all entries in the validation database. |
| `instance` | Clears validation database for a specific instance. |

Example: Clearing validation database options

```
clear validation database all
clear validation database instance vrf-blue
```

```
clear validation database instance vrf-blue 192.168.5.50 source 192.168.5.20
```

API To access the Operational State API that corresponds to this CLI, click include::common:partial$operational_api_links.adoc[tag=clear_validation_database].

# 2.6. IS-IS

## 2.6.1. IS-IS Overview

IS-IS, or Intermediate System to Intermediate System, is an open standard routing protocol. ISO published the standard as a way to route datagrams as part of their OSI stack. IETF later republished the standard, and added IP route support.

It is a link-state routing protocol, similar to OSPF. It forms neighbor adjacencies, has areas, exchanges link-state packets, builds a link-state database and runs the Dijkstra SPF algorithm to find the best path to each destination, which is installed in the routing table.

#### IS-IS Segment Routing

IS-IS in RBFS supports segment routing based on RFC 8667. IS-IS Segment Routing enhances the IS-IS protocol by introducing new TLVs to carry segment routing information within IS-IS protocol packets. It introduces Segment IDs (SIDs) to represent different types of segments, such as prefix SIDs (node SIDs and anycast SIDs) and adjacency SIDs.

> RFC and draft compliance are partial except as specified.

#### IS-IS Support Over Unnumbered Interfaces

An unnumbered interface is a point-to-point interface that is not explicitly configured with a dedicated IP address; instead, it borrows an IP address from a loopback interface. IS-IS can be set up over an unnumbered interface, enabling each router to use a loopback IP address to establish an IS-IS adjacency.

For details on configuring an unnumbered interface, refer to the Logical Interface Configuration section of the *Interfaces User Guide*.

##### IS-IS Route Leaking from Level 2 to Level 1

RBFS supports Level 2 to Level 1 route leaking in IS-IS. This feature allows Level 1 routers to learn specific Level 2 routes that are not automatically leaked to Level 1. By default, Level 1 routers reach other IS-IS levels (outside their area) by sending packets to the nearest Level 1/2 router via a default route.

In an IS-IS environment with LDP, Level 1 routers need to learn the loopback addresses of routers outside their area to establish LSPs (Label Switched Paths). Therefore, specific IP addresses must be leaked from Level 2 to Level 1 to provide required routing information. This feature allows a policy-based configuration on the Level 1/2 routers that connect the two levels. Route leaking selectively leaks particular Level 2 routes into Level 1 areas using predefined route policies.

#### Supported Platforms

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

### 2.6.2. IS-IS Configuration

#### Configuration Hierarchy

The diagram below illustrates the IS-IS configuration hierarchy.

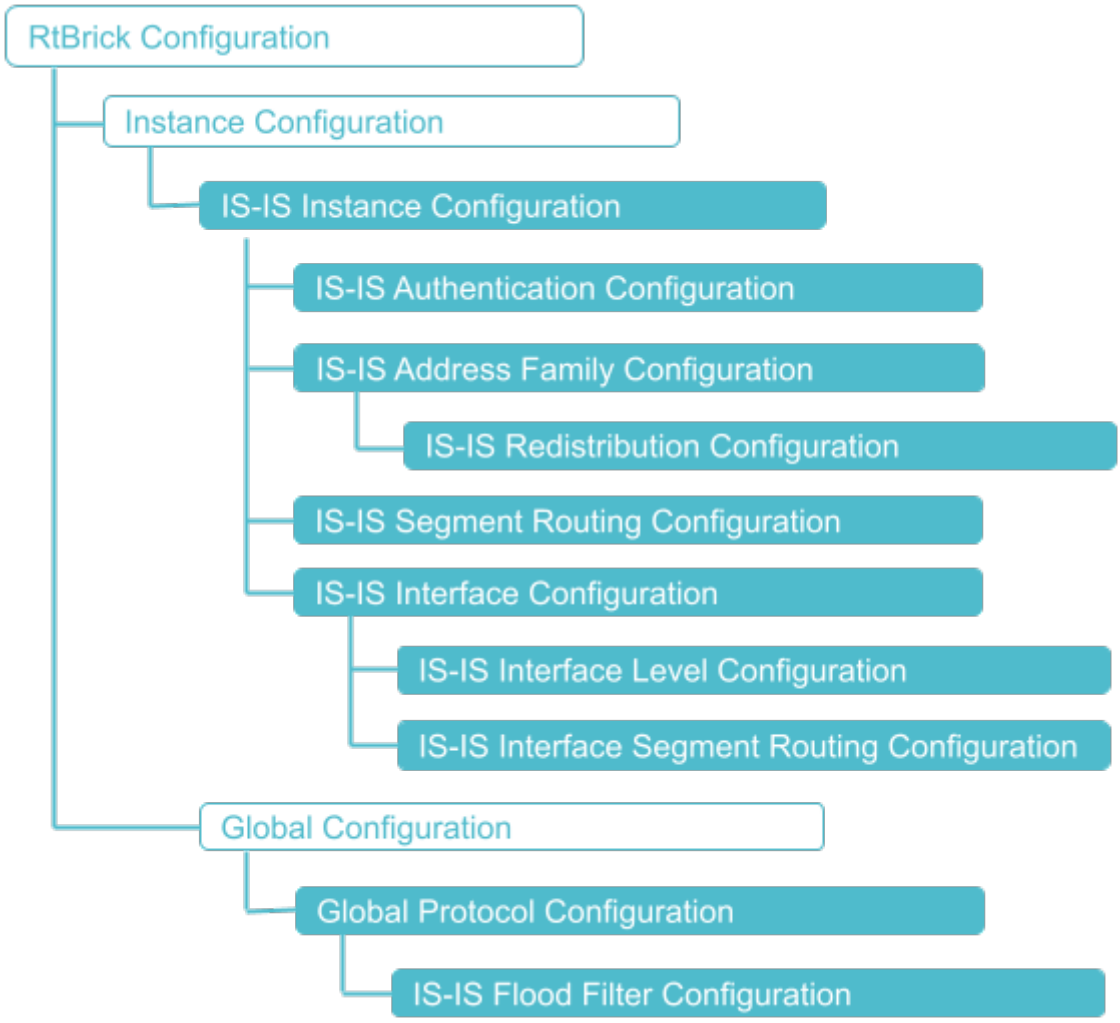



*Figure 14. IS-IS Configuration Hierarchy*

#### Configuration Syntax and Commands

The following sections describe the IS-IS configuration syntax and commands.

##### Instance Configuration

The instance configuration hierarchy includes parameters that are required for or used by IS-IS.

**Syntax:**

**set instance** <instance-name> **protocol isis** <attribute> <value>

| Attribute | Description |
|---|---|
| `<name>` | Name of the IS-IS instance |
| `area <area>` | IS-IS area-address. The area can be represented in 1, 3, 5, 13 bytes format. |
| `authentication <…>` | Specifies the authentication scheme for IS-IS. Refer to section IS-IS Authentication Configuration for the IS-IS authentication configuration details. |
| `holding-time <holding-time>` | Specifies how long a neighbor should consider this routing device to be operative without receiving another hello packet.<br><br>Default value: 30 seconds<br><br>Range: 3 to 180 seconds |
| `hostname <hostname>` | Specifies the hostname mapped to the system identifier. |
| `ignore-attached-bit [true/false]` | This configuration allows you to enable the routing device to ignore the attached bit on incoming Level 1 link-state PDUs. If the attached bit is ignored, no default route, which points to the routing device which has set the attached bit, is installed. Default: 'false'. |
| `interface <…>` | Name of the interface. Refer to section Configuring IS-IS Interface for the interface configuration details. |
| `ipv6-disable [true/false]` | Specifies whether the ipv6-disable configuration is enabled or not. When you set this value to "true", it indicates that IPv6 configuration is disabled. Default: False. |
| `[level-1/level-2] address-family <…>` | Protocol ISIS level-1/level-2 address-family configuration. Refer to section IS-IS Address-Family Configuration for the address family configuration details. |
| `level1-to-level2 route-leak [enable/disable]` | Specifies whether the level1-to-level2 route-leak is enabled or not. When set to disable, IS-IS will not leak routing information from a Level 1 area to a Level 2 area. By default, this option is enabled. |
| `lsp-lifetime <lsp-lifetime>` | IS-IS link-state PDUs maximum lifetime. Default value: 65535 seconds. Range: 317 - 65535. |

| Attribute | Description |
| --- | --- |
| `multipath <multipath>` | Load sharing among multiple IS-IS paths.<br><br>The default ECMP Maximum Next-hop Count is limited to 8; however, this limit varies depending on the platform that RBFS supports. For details on the limits set for the ECMP Maximum Next-hop Count on the platforms, refer to the Resource Limits/Feature Support section of the *Platform Guide*. |
| `no-mpls-transit-path [true/false]` | When set to true, IS-IS will not install segment routing transit path. Default: false |
| `overload [true/false]` | When set to true, IS-IS overload bit is set. Default: false |
| `router-id <router-id>` | ISIS router identifier (ipv4 format: A.B.C.D) |
| `system-id <system-id>` | Specifies the system ID of the device. |

Example: IS-IS Instance Configuration

The following configuration defines the instance 'default'.

```
supervisor@rtbrick: cfg>show config instance default protocol isis
{
  "rtbrick-config:isis": {
    "system-id": "1921.6800.1001",
    "area": [
      "49.0001/24"
      ],
    "hostname": "isr1",
 <...>
```

Example: Disabling IS-IS Route Leaking from a Level 1 Area to a Level 2 Area

The following configuration defines IS-IS route leaking behavior between Level 1 and Level 2 routing domains.

```
set instance default protocol isis level1-to-level2 route-leak disable
```

```
supervisor@rtbrick: cfg> show config instance default protocol isis level1-to-level2
{
  "rtbrick-config:level1-to-level2": {
    "route-leak": "disable"
  }
}
```

To access the RESTCONF API that corresponds to this CLI, click here.

#### IS-IS Authentication Configuration

**Syntax:**

```
set instance <instance-name> protocol isis authentication [level-1 | level-2] <attribute> <value>
```

| Attribute | Description |
|---|---|
| `check [disable \| enable]` | Specifies an authentication check to reject PDUs that do not match the type or key requirements. You can enable or disable the authentication check. |
| `key-id1 <key-id1> / key-id2 <key-id2>` | The key ID allows you to specify the key identifiers for level-1/level-2 authentication. |
| `key1-encrypted-text <key1-encrypted-text> / key2-encrypted-text <key2-encrypted-text>` | Authentication key1 and key 2 encrypted text |
| `key1-plain-text <key1-plain-text> / key2-plain-text <key2-plain-text>` | The level-1/level-2 authentication keys specify the authentication keys (passwords) that are used by the neighboring routing devices to verify the authenticity of packets sent from this interface. For the key to work, you also must include the authentication-type statement. |
| `type` | Enables you to specify the authentication type for IS-IS. If you enable authentication, you must specify a password by including the authentication-key statement.<br><br>The following authentication types are supported:<br><br>• clear_text: Simple password authentication.<br>• md5: Message-Digest Algorithm 5.<br>• sha1: Secure Hash Algorithm 1.<br>• sha224: Secure Hash Algorithm 224.<br>• sha256: Secure Hash Algorithm 256.<br>• sha384: Secure Hash Algorithm 384.<br>• sha512: Secure Hash Algorithm 512. |

Example configuration for IS-IS authentication.

The following configuration sets up authentication for IS-IS protocol at both level-1 and level-2. For both level-1 and level-2, the authentication type is set to 'md5'. It indicates that MD5 will be used to secure IS-IS routing updates. The authentication keys for level-1: key1-encrypted-text is set to "$24a6f10525a11077bec3b451be8855877" and for level-2 key1-encrypted-text is set to

"$2d787015bf84f3b58d5fd393a96c9639c". These key1-encrypted-text values are the encrypted (hashed) passwords used for authentication at each level.

```
set instance default protocol isis authentication level-1 type md5
set instance default protocol isis authentication level-1 key1-encrypted-text
"$24a6f10525a11077bec3b451be8855877"
set instance default protocol isis authentication level-2 type md5
set instance default protocol isis authentication level-2 key1-encrypted-text
"$2d787015bf84f3b58d5fd393a96c9639c"
```

```
supervisor@rtbrick: cfg> show config instance default protocol isis authentication
{
  "rtbrick-config:authentication": {
    "level-1": {
      "type": "md5",
      "key1-encrypted-text": "$24a6f10525a11077bec3b451be8855877"
    },
    "level-2": {
      "type": "md5",
      "key1-encrypted-text": "$2d787015bf84f3b58d5fd393a96c9639c"
    }
  }
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### IS-IS Address-Family Configuration

The address-family command allows you to enable the address families that IS-IS will route and configure settings that are specific to that address family.

**Syntax:**

**set instance** <instance-name> **protocol isis** [**level-1** | **level-2**] **address-family** <attribute> <value>

| Attribute | Description |
|---|---|
| <afi> | Address family identifier (AFI). Supported values: ipv4, ipv6 |
| <safi> | Subsequent address family identifier (SAFI). Supported values: unicast or labeled-unicast |

API To access the RESTCONF API that corresponds to this CLI (level-1), click here.

API To access the RESTCONF API that corresponds to this CLI (level-2), click here.

**Configuring Route Redistribution**

**Syntax:**

**set instance** <instance-name> **protocol isis** [**level-1** | **level-2**] **address-family** <afi> <safi> **redistribute** <attribute> <value>

| Attribute | Description |
|---|---|
| `<afi>` | Address family identifier (AFI). Supported values: ipv4, ipv6 |
| `<safi>` | Subsequent address family identifier (SAFI). Supported values: unicast or labeled-unicast |
| `redistribute <protocol>` | Specifies the source from which the routes are to be redistributed from. The available options include `arp-nd`, `bgp`, `bgp-local`, `bgp-local-origin`, `direct`, `igmp`, `ospf`, `l2tpv2`, `ldp`, `local`, `pim`, `ppp`, `rib`, and `static`. |
| `redistribute <protocol> <policy>` | Specifies the name of the policy map. The redistribute attach point allows routes from other sources to be advertised by IS-IS. Policy can be applied only to the routes that are redistributed from other sources to IS-IS. The support for inter-level leaking through policy is unavailable. |

Example: IS-IS address-family and route redistribution configuration

The following configuration defines how the IS-IS protocol at Level 1 will redistribute static routes for both IPv4 and IPv6 unicast address families. It sets up the IS-IS protocol within the "default" instance, specifically targeting Level 1 of the IS-IS hierarchy. For the IPv4 unicast address family, it specifies that routes learned from a static source should be redistributed. The policy applied during this redistribution is referred to as 'filter-link-address'.

Similarly, the configuration addresses the IPv6 unicast address family, redistributing routes from a static source. The policy applied is named 'filter-link6-address', which will be enforced during the redistribution of IPv6 routes.

```
set instance default protocol isis level-1 address-family ipv4
set instance default protocol isis level-1 address-family ipv4 safi unicast
set instance default protocol isis level-1 address-family ipv4 redistribute static
set instance default protocol isis level-1 address-family ipv4 redistribute static
policy filter-link-address
set instance default protocol isis level-1 address-family ipv6
set instance default protocol isis level-1 address-family ipv6 safi unicast
set instance default protocol isis level-1 address-family ipv6 redistribute static
set instance default protocol isis level-1 address-family ipv6 redistribute static
policy filter-link6-address
```

```
supervisor@rtbrick: cfg> show config instance default protocol isis level-1
```

```
{
  "rtbrick-config:level-1": {
    "address-family": [
      {
        "afi": "ipv4",
        "safi": "unicast",
        "redistribute": [
          {
            "source": "static",
            "policy": "filter-link-address"
          }
        ]
      },
      {
        "afi": "ipv6",
        "safi": "unicast",
        "redistribute": [
          {
            "source": "static",
            "policy": "filter-link6-address"
          }
        ]
      }
    ]
  }
}
```

API To access the RESTCONF API that corresponds to this CLI (level-1), click here.

API To access the RESTCONF API that corresponds to this CLI (level-2), click here.

#### Segment Routing Configuration

IS-IS segment routing extensions allow to advertise labels with prefixes.

RBFS currently supports the following IS-IS segment routing features:

- MPLS data plane
- IPv4 prefixes (TLV 135) and IPv6 prefixes (TLV 236)
- Prefix SID with node flag (Node SID) on loopback interface
- Anycast SID
- A single global SRGB block
- Adjacency SIDs

**Syntax:**

**set instance** <instance-name> **protocol isis segment-routing** <attribute> <value>

| Attribute | Description |
|---|---|
| `srgb base <srgb base>' | Specifies the segment routing global block (SRGB) in source packet routing. SRGB is used for prefix SIDs.<br>Supported MPLS label values range: 256 - 499968 In RBFS, BGP uses the label range 20000 - 100000. It is recommended to assign label values outside of these reserved ranges to avoid conflicts. |
| `srgb range <srgb range>` | IS-IS system range of labels from the base label. Range: 0 - 499968.<br>For details on the supported label ranges, see the Static, System, and Dynamic Label Ranges section in the Platform Guide. |
| `srlb base <srlb base>` | Specifies the segment routing local block (SRLB) in source packet routing. SRLB is used for adjacency SIDs.<br>Supported MPLS label value range: 256 - 499968. In RBFS, BGP uses the label range 20000 - 100000. It is recommended to assign label values outside of these reserved ranges to avoid conflicts. |
| `srlb range <srlb range>` | IS-IS system range of labels from the base label. Range: 0 - 499968.<br>For details on the supported label ranges, see the Static, System, and Dynamic Label Ranges section in the Platform Guide. |

Example: IS-IS Segment Routing Configuration

This configuration establishes segment routing for the IS-IS instance. It defines two Segment Routing Global Blocks (SRGBs) and two Segment Routing Label Blocks (SRLBs). The SRLB has a base value of 2000 and a range of 1000, which means it will use labels from 2000 to 2999. Similarly, the SRGB has a base value of 1000 and a range of 1000, allowing it to use labels from 1000 to 1999. These values are essential for assigning Segment Identifiers (SIDs) within a Segment Routing domain.

```
set instance default protocol isis segment-routing srlb base 2000
set instance default protocol isis segment-routing srlb range 1000
set instance default protocol isis segment-routing srgb base 1000
set instance default protocol isis segment-routing srgb range 1000
```

```
supervisor@rtbrick>spine1: cfg> show config instance default protocol isis segment-
routing
{
  "rtbrick-config:segment-routing": {
    "srlb": {
      "base": 2000,
      "range": 1000
```

```
      },
      "srgb": {
        "base": 1000,
        "range": 1000
      }
    }
  }
```

To access the RESTCONF API that corresponds to this CLI, click here.

#### Configuring IS-IS Interface

By default, there are no interfaces associated with IS-IS. You must configure at least one IS-IS interface for IS-IS adjacency formation.

**Syntax:**

**set instance** <instance> **protocol isis interface** <name> <attribute> <value>

| Attribute | Description |
|---|---|
| `<name>` | Specifies the name of the IS-IS interface. IS-IS can be configured on both numbered and unnumbered interfaces. |
| `flood-filter <flood-filter>` | Specifies the IS-IS flood filter name |
| `hello-interval` | Specifies IS-IS system interface hello interval. Range: 1 - 600, Default :10s.<br>To configure hello-interval, the user can either configure holding-time/hello-interval under interface or hello-interval under level [broadcast interface only].<br>The order of priority, from highest to lowest, is as follows:<br>* For Broadcast interface type:<br>hello-interval under interface level 1/2 > hello-interval under interface > holding-time under instance<br>* For Point-to-Point interface type:<br>hello-interval under interface > holding-time under instance |
| `level-1 / level-2` | Specify IS-IS interface level configuration. Refer to section IS-IS Interface Level Configuration for the IS-IS interface level configuration details. |
| `lsp-interval <lsp-interval>` | IS-IS system interface LSP interval. Default value: 100. Range: 1 - 1000. |

| Attribute | Description |
|---|---|
| `passive [true / false]` | Enable interface in passive mode. Default: false |
| `system-id <system-id>` | Interface level system id |
| `type [broadcast / point-to-point]` | Specifies the type of the IS-IS system interface. Default: point-to-point.<br><br>broadcast—Specifies a broadcast (or LAN) interface.<br>point-to-point—Specifies a point-to-point interface. |
| `ldp-synchronization [enable / disable]` | Enable LDP IGP synchronization. Default: disable |
| `segment-routing <afi> no-penultimate-pop <true \| false>` | Enables (true) or disables (false) no-penultimate-hop-pop. Default: false<br>The “no-penultimate-pop” configuration ensures that the second-to-last router (penultimate hop) in a Label Switched Path (LSP) does not remove (or does not "pop") the MPLS label from a packet before forwarding it to the egress router. |
| `segment-routing <afi> explicit-null <true \| false>` | Enables (true) or disables (false) explicit-null. Default: false.<br>The "explicit-null" configuration assigns an explicit null label (0) for IPv4 and an explicit null label (2) for IPv6 to the penultimate router. The “no-penultimate-pop” feature must be configured to enable “explicit-null”. |

Example: IS-IS Interface Configuration

The following configuration defines the IS-IS protocol settings for three interfaces: `ifl-0/0/2/0`, `ifl-0/0/2/1`, and `lo-0/0/0/0`. It specifies parameters such as interface types (point-to-point and broadcast), authentication settings, LSP (Link State Protocol) intervals, LDP (Label Distribution Protocol) synchronization, metrics, priorities, and hello intervals for both Level-1 and Level-2 IS-IS on each of these interfaces. Also, it configures Segment Routing with IPv4 and IPv6 SID (Segment Identifier) indices for the loopback interface.

For the interface `ifl-0/0/2/0`, the type is set to `point-to-point`, indicating that this interface is configured for a direct, one-to-one connection. The passive setting is set to `false`, meaning IS-IS will send and receive hello messages on this interface. Both SNP (Synchronized Network Protocol) and hello authentication for Level-1 are disabled.

For the interface `ifl-0/0/2/1`, a flood filter named `spine1_lsr1_flood_filter` is applied to control the flooding of IS-IS LSPs. This interface type is defined as `broadcast`, indicating that it is a multi-access interface. The LSP interface interval is set to `200`, which determines the time interval at which LSPs are generated. LDP synchronization is enabled that allows synchronization between IS-IS and LDP. For the interface `lo-0/0/0/0`, the passive setting is true, which means IS-IS will not send hello messages or LSPs on this interface.

```
set instance default protocol isis interface ifl-0/0/2/0 passive false
set instance default protocol isis interface ifl-0/0/2/0 level-1 snp-authentication
disable
```

```
set instance default protocol isis interface ifl-0/0/2/0 level-1 hello-authentication
disable
set instance default protocol isis interface ifl-0/0/2/1 flood-filter
spine1_lsr1_flood_filter
set instance default protocol isis interface ifl-0/0/2/1 type broadcast
set instance default protocol isis interface ifl-0/0/2/0 type point-to-point
set instance default protocol isis interface ifl-0/0/2/1 lsp-interval 200
set instance default protocol isis interface ifl-0/0/2/1 ldp-synchronization enable
set instance default protocol isis interface ifl-0/0/2/1 level-1 snp-authentication
enable
set instance default protocol isis interface ifl-0/0/2/1 level-1 hello-authentication
enable
set instance default protocol isis interface ifl-0/0/2/1 level-1 metric 1000
set instance default protocol isis interface ifl-0/0/2/1 level-1 adjacency-disable
false
set instance default protocol isis interface ifl-0/0/2/1 level-1 priority 100
set instance default protocol isis interface ifl-0/0/2/1 level-1 hello-interval 20
set instance default protocol isis interface ifl-0/0/2/1 level-2 priority 100
set instance default protocol isis interface ifl-0/0/2/1 level-2 hello-interval 20
set instance default protocol isis interface lo-0/0/0/0 passive true
set instance default protocol isis interface lo-0/0/0/0 segment-routing ipv4 index 100
set instance default protocol isis interface lo-0/0/0/0 segment-routing ipv6 index 102
```

```
supervisor@rtbrick>spine1: cfg> show config instance default protocol isis interface
{
  "rtbrick-config:interface": [
    {
      "name": "ifl-0/0/2/0",
      "type": "point-to-point",
      "passive": "false",
      "level-1": {
        "snp-authentication": "disable",
        "hello-authentication": "disable"
      },
      "level-2": {}
    },
    {
      "name": "ifl-0/0/2/1",
      "flood-filter": "spine1_lsr1_flood_filter",
      "type": "broadcast",
      "lsp-interval": 200,
      "ldp-synchronization": "enable",
      "level-1": {
        "snp-authentication": "enable",
        "hello-authentication": "enable",
        "metric": 1000,
        "adjacency-disable": "false",
        "priority": 100,
        "hello-interval": 20
      },
```

```
      "level-2": {
        "priority": 100,
        "hello-interval": 20
      }
    },
    {
      "name": "lo-0/0/0/0",
      "passive": "true",
      "segment-routing": {
        "ipv4": {
          "index": 100
        },
        "ipv6": {
          "index": 102
        }
      },
      "level-1": {},
      "level-2": {}
    }
  ]
}
```

Example: IS-IS Interface Level Flood Filter Configuration

This configuration sets a flood filter named 'spine1_lsr1_flood_filter' on the interface 'ifl-0/0/2/1' for S-IS protocol to control the flooding of routing updates.

```
set instance default protocol isis interface ifl-0/0/2/1 flood-filter
spine1_lsr1_flood_filter
```

```
supervisor@rtbrick>spine1: cfg> show config instance default protocol isis interface
ifl-0/0/2/1 flood-filter
{
  "rtbrick-config:flood-filter": "spine1_lsr1_flood_filter"
}
```

Example: IS-IS Interface Configuration with enabled LDP synchronization

```
set instance default protocol isis interface ifl-0/0/2/1 ldp-synchronization enable
```

```
supervisor@rtbrick: cfg> show config instance default protocol isis interface ifl-
0/0/2/1 ldp-synchronization
{
  "rtbrick-config:ldp-synchronization": "enable"
}
```

Example: IS-IS Interface Configuration for a Broadcast Interface.

This configuration enables LDP synchronization on the interface 'ifl-0/0/2/1' for the default instance for the IS-IS routing protocol.

```
set instance default protocol isis interface ifl-0/0/2/1 ldp-synchronization enable
```

```
supervisor@rtbrick>spine1: cfg> show config instance default protocol isis interface
ifl-0/0/2/1 type
{
  "rtbrick-config:type": "broadcast"
}
```

Example: P/E Flags Configuration for SID in IS-IS Segment Routing

```
set instance default protocol isis interface lo-0/0/0/1 segment-routing ipv4 no-
penultimate-pop true
```

The following command configures to assign an explicit null label for the penultimate router on the egress LSP. The configuration sets the 'lo-0/0/0/0' interface for the default IS-IS protocol instance. This interface is configured as passive for IS-IS and enables Segment Routing for both IPv4 and IPv6. The configuration includes assigning Segment Identifiers, enabling 'no-penultimate-pop' behavior, and advertising explicit null labels for both IPv4 and IPv6.

```
set instance default protocol isis interface lo-0/0/0/0 passive true
set instance default protocol isis interface lo-0/0/0/0 segment-routing ipv4 index 100
set instance default protocol isis interface lo-0/0/0/0 segment-routing ipv4 no-
penultimate-pop true
set instance default protocol isis interface lo-0/0/0/0 segment-routing ipv4 explicit-
null true
set instance default protocol isis interface lo-0/0/0/0 segment-routing ipv6 index 106
Set instance default protocol isis interface lo-0/0/0/0 segment-routing ipv6 no-
penultimate-pop true
set instance default protocol isis interface lo-0/0/0/0 segment-routing ipv6 explicit-
null true
```

```
supervisor@rtbrick.net: cfg> show config instance default protocol isis interface lo-
0/0/0/0
{
  "rtbrick-config:interface": [
    {
      "name": "lo-0/0/0/0",
      "passive": "true",
      "segment-routing": {
        "ipv4": {
```

```
          "index": 100,
          "no-penultimate-pop": "true",
          "explicit-null": "true"
        },
        "ipv6": {
          "index": 106,
          "no-penultimate-pop": "true",
          "explicit-null": "true"
        }
      }
    }
  ]
}
```

To access the RESTCONF API that corresponds to this CLI, click here.

#### IS-IS Interface Level Configuration

**Syntax:**

**set instance** <instance> **protocol isis interface** <name> [**level-1** | **level-2**] <attribute> <value>

| Attribute | Description |
|---|---|
| `adjacency-disable [true/false]` | Specify the level-1/level-2 adjacency on an interface. Default: false. |
| `hello-authentication [disable/enable]` | Authentication on hello packets. |
| `hello-interval` | Specifies the length of time between the sending of IS-IS hello PDUs. Default: 10. The hello interval can be set for both `broadcast` and `point-to-point` interfaces that are configured for Levels 1 and 2. |
| `priority` | Specify the priority on a broadcast interface. Default: 64. |
| `metric <metric>` | Level-1/Level-2 metric on an interface. Default: 1000000. |
| `snp-authentication [enable/disable]` | Authentication on CSNP/PSNP packets. |

Example: IS-IS Interface Level Configuration

This configuration defines the parameters for the IS-IS interface 'ifl-0/0/2/1'. It includes various Level 1 and Level 2 parameters such as authentication for SNPs and hello messages, metrics, priorities, and hello intervals. It also defines the broadcast type, LSP intervals, and LDP synchronization.

For Level 1, `snp-authentication` and `hello-authentication` are enabled. The `metric` is set to 1000 or determining optimal paths. The `adjacency-disable` option is set to 'false', allowing Level 1

adjacencies. The `priority` is set to 100, and the `hello-interval` defines the interval for sending Level 1 Hello PDUs as 20 seconds.

For Level 2, the `priority` is also set to 100, influencing the Level 2 Designated Intermediate System (DIS) election, and the `hello-interval` is similarly set to 20 seconds for sending Level 2 Hello PDUs.

```
set instance default protocol isis interface ifl-0/0/2/1 flood-filter
spine1_lsr1_flood_filter
set instance default protocol isis interface ifl-0/0/2/1 type broadcast
set instance default protocol isis interface ifl-0/0/2/1 lsp-interval 200
set instance default protocol isis interface ifl-0/0/2/1 ldp-synchronization enable
set instance default protocol isis interface ifl-0/0/2/1 level-1 snp-authentication
enable
set instance default protocol isis interface ifl-0/0/2/1 level-1 hello-authentication
enable
set instance default protocol isis interface ifl-0/0/2/1 level-1 metric 1000
set instance default protocol isis interface ifl-0/0/2/1 level-1 adjacency-disable
false
set instance default protocol isis interface ifl-0/0/2/1 level-1 priority 100
set instance default protocol isis interface ifl-0/0/2/1 level-1 hello-interval 20
set instance default protocol isis interface ifl-0/0/2/1 level-2 priority 100
set instance default protocol isis interface ifl-0/0/2/1 level-2 hello-interval 20
```

```
supervisor@rtbrick>spine1: cfg> show config instance default protocol isis interface
ifl-0/0/2/1
{
  "rtbrick-config:interface": [
    {
      "name": "ifl-0/0/2/1",
      "flood-filter": "spine1_lsr1_flood_filter",
      "type": "broadcast",
      "lsp-interval": 200,
      "ldp-synchronization": "enable",
      "level-1": {
        "snp-authentication": "enable",
        "hello-authentication": "enable",
        "metric": 1000,
        "adjacency-disable": "false",
        "priority": 100,
        "hello-interval": 20
      },
      "level-2": {
        "priority": 100,
        "hello-interval": 20
      }
    }
  ]
}
```

To access the RESTCONF API that corresponds to this CLI, click here.

#### Interface-level Segment Routing Configuration

**Syntax:**

**set instance** <instance> **protocol isis interface** <name> **segment-routing** <attribute> <value>

| Attribute | Description |
|---|---|
| `segment-routing [ipv4 / ipv6] anycast-index <anycast-index>` | Anycast index segment-ID. The prefix SIDs and anycast SIDs are applied on loopback interface only. |
| `segment-routing [ipv4 / ipv6] index <index>` | Prefix index segment ID. |
| `segment-routing point-to-point [ipv4 / ipv6] adjacency-index <adjacency-index>` | Adjacency index segment-ID. The adjacency SIDs are applied on active IS-IS interfaces on which adjacencies are established. |

Example: IS-IS Interface Level Segment Routing Configuration for Prefix and Anycast SID

This configuration establishes IS-IS Interface Level Segment Routing for both IPv4 and IPv6 on two loopback interfaces. The interface 'lo-0/0/0' is assigned unique Segment Identifier indices: 100 for IPv4 and 200 for IPv6. These SIDs are utilized for prefix routing, allowing for the identification of specific network segments. Also, the loopback interface 'lo-0/0/1' is configured with anycast SID indices of 110 for IPv4 and 210 for IPv6. This configuration enables segment routing on both of the interfaces.

```
set instance default protocol isis interface lo-0/0/0
set instance default protocol isis interface lo-0/0/0 segment-routing ipv4
set instance default protocol isis interface lo-0/0/0 segment-routing ipv4 index 100
set instance default protocol isis interface lo-0/0/0 segment-routing ipv6
set instance default protocol isis interface lo-0/0/0 segment-routing ipv6 index 200
set instance default protocol isis interface lo-0/0/1 segment-routing ipv4 anycast-
index 110
set instance default protocol isis interface lo-0/0/1 segment-routing ipv6 anycast-
index 210
```

```
supervisor@rtbrick>spine1: cfg> show config instance default protocol isis interface
{
  "rtbrick-config:instance": [
      {
        "name": "default",
        "protocol": {
          "isis": {
            "interface": [
              {
                "name": "lo-0/0/0",
```

```
                "segment-routing": {
                  "ipv4": {
                    "index": 100
                  },
                  "ipv6": {
                    "index": 200
                  }
                }
              },
              {
                "name": "lo-0/0/1",
                "segment-routing": {
                  "ipv4": {
                    "anycast-index": 110
                  },
                  "ipv6": {
                    "anycast-index": 210
                  }
                }
              }
            ]
          }
        }
      }
    ]
  }
}
```


Example: IS-IS Interface Level Segment Routing Configuration for Adjacency SID

This configuration establishes IS-IS Interface Level Segment Routing for both IPv4 and IPv6 on two loopback interfaces. The interface 'lo-0/0/0' is assigned unique Segment Identifier indices: 100 for IPv4 and 200 for IPv6. These SIDs are utilized for prefix routing. Also, the loopback interface 'lo-0/0/1' is configured with anycast SID indices '110', for IPv4 and 210 for IPv6. It enables segment routing on both of the interfaces.

```
set instance default protocol isis interface ifl-0/0/2/0
set instance default protocol isis interface ifl-0/0/2/0 segment-routing point-to-
point ipv4
set instance default protocol isis interface ifl-0/0/2/0 segment-routing point-to-
point ipv4 adjacency-index 111
set instance default protocol isis interface ifl-0/0/2/0 segment-routing point-to-
point ipv6
set instance default protocol isis interface ifl-0/0/2/0 segment-routing point-to-
point ipv6 adjacency-index 112
```

```
supervisor@rtbrick>spine1: cfg> show config instance default protocol isis interface
ifl-0/0/2/0 segment-routing
```

```
{
  "rtbrick-config:segment-routing": {
    "point-to-point": {
      "ipv4": {
        "adjacency-index": 111
      },
      "ipv6": {
        "adjacency-index": 112
      }
    }
  }
}
```

To access the RESTCONF API that corresponds to this CLI, click here.

#### IS-IS Global Configuration

#### IS-IS Flood Filter Configuration

In IS-IS, by default all routers flood link-state packets, so that all routers will have a complete topology view. IS-IS flood filters allow to modify this behavior and limit the exchange of LSPs. For example, if two spine routers in a spine/leaf fabric are symmetrically connected to two upstream label-switch routers (LSR) like shown in the figure below, you can use a flood filter to not advertise LSPs learned from LSR A back to the LSR B via the second spine switch.

The flooding filter configuration is part of the global configuration hierarchy and therefore you can configure filtering globally, i.e. not per instance, so that the filter configurations can be reused across instances.

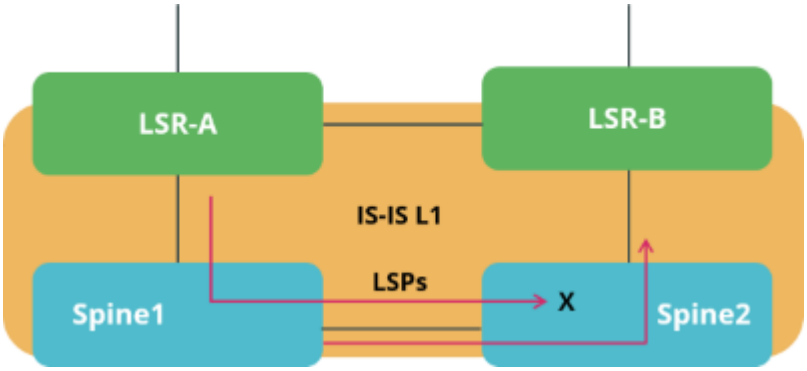



*Figure 15. IS-IS Flood Filter Configuration*

**Syntax:**

**set global protocol isis flood-filter** <filter-name> <ordinal> <attribute> <value>

| Attribute | Description |
|---|---|
| `<filter-name>` | Filter-name which binds a flooding filter to an IS-IS interface |
| `<ordinal>` | Number to filter rule |

| Attribute | Description |
|---|---|
| `action [block/flood]` | Action required to flood or not |
| `ordinal-name <ordinal-name>` | Name for the filter rule |
| `system-id <system-id>` | IS-IS instance system-id |
| `system-id-mask <system-id-mask>` | System ID mask on which the filter should match |

Example: IS-IS Flood Filter Configuration

This configuration sets up a flood filter named 'spine1_lsr1_flood_filter' for the interface 'ifl-0/0/2/1'.

```
set instance default protocol isis interface ifl-0/0/2/1 flood-filter flood-filter
spine1_lsr1_flood_filter
```

```
supervisor@rtbrick>spine1: cfg> show config instance default protocol isis interface
ifl-0/0/2/1 flood-filter
{
  "rtbrick-config:flood-filter": "spine1_lsr1_flood_filter"
}
```

#### Level 2 to Level 1 Policy Configuration

The policy configuration allows to define a policy for leaking all the level 2 routes to level 1.

For more information about Policy configuration, see the Policy User Guide.

**Syntax:**

**set policy statement isis_l2tol1_leak ordinal** <value> **action rule** <value> **operation return-permit**

| Attribute | Description |
|---|---|
| ordinal <value> | Specifies the ordinal value. |
| rule <number> | Specify the action rule number. |
| operation return-permit | Stops policy execution and return result as permit. |

The following configuration defines a routing policy named 'isis_l2tol1_leak' that permits all routes to be leaked from IS-IS Level 2 into Level 1. The policy contains a rule (ordinal 1) with an action to 'return-permit'. Since no specific prefixes are matched, the permit action applies to all routes by default. This configuration allows all Level 2 routes to be leaked into Level 1 routers.

```
set policy statement isis_l2tol1_leak
set policy statement isis_l2tol1_leak ordinal 1
```

```
set policy statement isis_l2tol1_leak ordinal 1 action
set policy statement isis_l2tol1_leak ordinal 1 action rule 1
set policy statement isis_l2tol1_leak ordinal 1 action rule 1 operation return-permit
```

```
supervisor@rtbrick.net: cfg> show config policy
{
  "rtbrick-config:policy": {
    "statement": [
      {
        "name": "isis_l2tol1_leak",
        "ordinal": [
          {
            "ordinal": 1,
            "action": {
              "rule": [
                {
                  "rule": 1,
                  "operation": "return-permit"
                }
              ]
            }
          }
        ]
      }
    ]
  }
}
```

#### Level 2 to Level 1 Policy Configuration for a Specific IPv4 Route

The policy configuration allows to define a policy for leaking a specific IPv4 level 2 routes to level 1.

For more information about Policy configuration, see the Policy User Guide.

**Syntax**:

**set policy statement** <policy-name> **ordinal** <number> **match rule** <number> <attribute> <value>

| Attribute | Description |
|---|---|
| `rule <number>` | Specifies the match rule number. |
| `type <attribute-type>` | Specifies the attribute type. |
| `match-type <match-type>` | Specifies the match type. |
| `value <value>` | Attribute value. This is the actual value of the attribute to match, for example an IP prefix. |

| Attribute | Description |
| --- | --- |
| `value-type <value-type>` | Attribute value type. |

The following policy configuration allows only a specific IPv4 route to be leaked from level2 to level1 and blocking the other level2 route leaking to level1. It defines a routing policy named 'isis_l2tol1_leak' that controls which IS-IS Level 2 routes are leaked into Level 1. The rule 'ordinal 1' matches the exact IPv4 prefix 192.168.1.3/32 and allows it to be leaked from Level 2 to Level 1. The second rule 'ordinal 10' is as a default deny (return-deny), preventing any other prefixes from being leaked. This ensures only the explicitly defined route is distributed to Level 1 routers. The policy contains rules to leak a specific route to Level 1 routers and to deny all other routes from leaking.

```
set policy statement isis_l2tol1_leak
set policy statement isis_l2tol1_leak ordinal 1
set policy statement isis_l2tol1_leak ordinal 1 match
set policy statement isis_l2tol1_leak ordinal 1 match rule 1
set policy statement isis_l2tol1_leak ordinal 1 match rule 1 type ipv4-prefix
set policy statement isis_l2tol1_leak ordinal 1 match rule 1 value-type discrete
set policy statement isis_l2tol1_leak ordinal 1 match rule 1 match-type exact
set policy statement isis_l2tol1_leak ordinal 1 match rule 1 value 192.168.1.3/32
set policy statement isis_l2tol1_leak ordinal 1 action
set policy statement isis_l2tol1_leak ordinal 1 action rule 1
set policy statement isis_l2tol1_leak ordinal 1 action rule 1 operation return-permit
set policy statement isis_l2tol1_leak ordinal 10
set policy statement isis_l2tol1_leak ordinal 10 action
set policy statement isis_l2tol1_leak ordinal 10 action rule 1
set policy statement isis_l2tol1_leak ordinal 10 action rule 1 operation return-deny
```

```
supervisor@rtbrick.net: cfg> show config policy
{
  "rtbrick-config:policy": {
    "statement": [
      {
        "name": "isis_l2tol1_leak",
        "ordinal": [
          {
            "ordinal": 1,
            "match": {
              "rule": [
                {
                  "rule": 1,
                  "type": "ipv4-prefix",
                  "value-type": "discrete",
                  "match-type": "exact",
                  "value": [
                    "192.168.1.3/32"
                    ]
                }
              ]
```

```
            },
            "action": {
              "rule": [
                {
                  "rule": 1,
                  "operation": "return-permit"
                }
              ]
            }
          },
          {
            "ordinal": 10,
            "action": {
              "rule": [
                {
                  "rule": 1,
                  "operation": "return-deny"
                }
              ]
            }
          }
        ]
      }
    ]
  }
}
```

**IS-IS Level 2 to Level 1 Route Leaking Configuration**

This configuration associates level2 to level1 route leaking policy with the IS-IS instance.

> Level 2 to Level 1 route leaking policy must be created prior to referencing it to the IS-IS instance.

**Syntax:**

**set instance** <instance-name> **protocol isis level2-to-level1 route-leak policy** <name>

| Attribute | Description |
|---|---|
| `<instance-name>` | Specifies the name of the instance. |
| `policy-name` | Specify name of the route leak policy. |

Example: IS-IS Level 2 to Level 1 Route Leaking Configuration

This following configuration enables route leaking from IS-IS Level 2 to Level 1 within the default routing instance. It allows specific Level 2 routes to be advertised into the Level 1 area using the defined route policy 'isis_l2tol1_leak'. The policy determines which routes are eligible for leaking

and provides controlled redistribution between IS-IS levels for improved reachability.

```
set instance default protocol isis level2-to-level1 route-leak policy isis_l2tol1_leak
```

```
supervisor@rtbrick.net: cfg> show config instance default protocol isis level2-to-
level1
{
  "rtbrick-config:level2-to-level1": {
    "route-leak": {
      "policy": [
        "isis_l2tol1_leak"
        ]
    }
  }
}
```

### 2.6.3. IS-IS Operational Commands

#### IS-IS Show Commands

The IS-IS show commands provide detailed information about the IS-IS protocol operation and IS-IS routes.

##### IS-IS Overview

**Syntax:**

**show isis overview**

| Option | Description |
|---|---|
| - | Without any option, this command displays a summary of all the IS-IS instances |

Example: Summary view of all the IS-IS instances

```
supervisor@rtbrick: op> show isis overview
Instance: default
  System ID: 1980.5110.0111
  System hostname: isr1
  Areas: 49.0001/24
  Neighbor hold time: 30 sec
  LSP life time: 65535 sec
  Overload bit set: False
  SRGB base: 10000
  SRGB range: 1000
  SRGB label values: 10000 - 10999
  SRLB base: 20000
```

```
  SRLB range: 1000
  SRLB label values: 20000 - 20999
  Authentication:
    Level 1: md5, Level 2: md5
```

API To access the Operational State API that corresponds to this CLI, click include::common:partial$operational_api_links.adoc[tag=show_isis_overview].

#### IS-IS Interface

**Syntax:**

**show isis interface** <option>

| Option | Description |
|---|---|
| - | Without any option, this command displays a summary of all the IS-IS interfaces |
| instance | Displays IS-IS interface information for an instance |
| statistics | Displays IS-IS interface statistics information |
| detail | Displays detailed output for all interfaces. |

Example 1: Summary view of the IS-IS interfaces

```
supervisor@rtbrick: op> show isis interface
Instance: default
  Interface            Level   Adjacencies    Metric     Type                 Passive
Unnumbered  LDP-sync
  lo-0/0/0/0               1             0    1000000    loopback                   1
0  -
  lo-0/0/0/0               2             0    1000000    loopback                   1
0  -
  lo-0/0/0/1               1             0    1000000    loopback                   1
0  -
  lo-0/0/0/1               2             0    1000000    loopback                   1
0  -
  ifl-0/0/2/0              1             1    1000000    point-to-point             0
0  -
  ifl-0/0/2/0              2             1    1000000    point-to-point             0
0  -
  ifl-0/0/2/100            1             1    1000000    broadcast                  0
0  -
  ifl-0/0/2/100            2             1    1000000    broadcast                  0
0  -
```

Example 2: Summary view of the IS-IS interfaces for a specific instance

```
supervisor@rtbrick: op> show isis interface instance default
Instance: default
  Interface            Level   Adjacencies    Metric     Type                 Passive
Unnumbered  LDP-sync
  lo-0/0/0/0               1             0    1000000    loopback                   1
0  -
  lo-0/0/0/0               2             0    1000000    loopback                   1
0  -
  lo-0/0/0/1               1             0    1000000    loopback                   1
0  -
  lo-0/0/0/1               2             0    1000000    loopback                   1
0  -
  ifl-0/0/2/0              1             1    1000000    point-to-point             0
0  -
  ifl-0/0/2/0              2             1    1000000    point-to-point             0
0  -
  ifl-0/0/2/100            1             1    1000000    broadcast                  0
0  -
  ifl-0/0/2/100            2             1    1000000    broadcast                  0
0  -
```

Example 3: Summary view of the IS-IS interfaces for a specific interface

```
supervisor@rtbrick: op> show isis interface ifl-0/0/2/0
Instance: default
  Interface: ifl-0/0/2/0, Level: 1
    Type: point-to-point, Passive: False, Unnumbered: False
    Metric: 1000000
    Hello interval: 10 seconds, Hello multiplier: 3, Hold time: 30 seconds
    Adjacencies: 1
    CNSP in: 2, CNSP out: 3, CNSP success: 2, CNSP fail: 0
    PSNP in: 1, PSNP out: 1, PSNP success: 1, PSNP fail: 0
    LSP in: 2, LSP out: 0, LSP success: 2, LSP fail: 0, LSP in purge: 0 LSP in auth
fail: 0
    IIH in: 4, IIH out: 5
  Interface: ifl-0/0/2/0, Level: 2
    Type: point-to-point, Passive: False, Unnumbered: False
    Metric: 1000000
    Hello interval: 10 seconds, Hello multiplier: 3, Hold time: 30 seconds
    Adjacencies: 1
    CNSP in: 2, CNSP out: 3, CNSP success: 2, CNSP fail: 0
    PSNP in: 3, PSNP out: 3, PSNP success: 3, PSNP fail: 0
    LSP in: 4, LSP out: 0, LSP success: 4, LSP fail: 0, LSP in purge: 0 LSP in auth
fail: 0
    IIH in: 4, IIH out: 5
```

Example 4: Summary view of the IS-IS interface statistics

```
supervisor@rtbrick: op> show isis interface statistics
Instance: default
  Interface             Level  CSNP In  CSNP Out  CSNP Fail  PSNP In  PSNP Out  PSNP
Fail  LSP In  LSP Out  LSP Fail  IIH In  IIH Out
  ifl-0/0/2/0               1        2         3          0        1         1
0       2        0         0       5        5
  ifl-0/0/2/0               2        2         3          0        3         3
0       4        0         0       5        5
  ifl-0/0/2/100             1        2         0          0        0         0
0       0        0         0      12       12
  ifl-0/0/2/100             2        2         0          0        0         0
0       0        0         0      12       12
```

Example 5: Detailed output of the IS-IS interface

```
supervisor@rtbrick: op> show isis interface detail
Instance: default
  Interface: lo-0/0/0/0, Level: 1
    Type: loopback, Passive: True, Unnumbered: False
    Metric: 1000000
  Interface: lo-0/0/0/0, Level: 2
    Type: loopback, Passive: True, Unnumbered: False
    Metric: 1000000
Instance: default
  Interface: lo-0/0/0/1, Level: 1
    Type: loopback, Passive: True, Unnumbered: False
    Metric: 1000000
  Interface: lo-0/0/0/1, Level: 2
    Type: loopback, Passive: True, Unnumbered: False
    Metric: 1000000
Instance: default
  Interface: ifl-0/0/2/0, Level: 1
    Type: point-to-point, Passive: False, Unnumbered: False
    Metric: 1000000
    Hello interval: 10 seconds, Hello multiplier: 3, Hold time: 30 seconds
    Adjacencies: 1
    CNSP in: 2, CNSP out: 3, CNSP success: 2, CNSP fail: 0
    PSNP in: 1, PSNP out: 1, PSNP success: 1, PSNP fail: 0
    LSP in: 2, LSP out: 0, LSP success: 2, LSP fail: 0, LSP in purge: 0 LSP in auth
fail: 0
    IIH in: 5, IIH out: 5
  Interface: ifl-0/0/2/0, Level: 2
    Type: point-to-point, Passive: False, Unnumbered: False
    Metric: 1000000
    Hello interval: 10 seconds, Hello multiplier: 3, Hold time: 30 seconds
    Adjacencies: 1
    CNSP in: 2, CNSP out: 3, CNSP success: 2, CNSP fail: 0
    PSNP in: 3, PSNP out: 3, PSNP success: 3, PSNP fail: 0
    LSP in: 4, LSP out: 0, LSP success: 4, LSP fail: 0, LSP in purge: 0 LSP in auth
```

```
fail: 0
    IIH in: 5, IIH out: 5
Instance: default
  Interface: ifl-0/0/2/100, Level: 1
    Type: broadcast, Passive: False, Unnumbered: False, Circuit ID: 0x1
    Metric: 1000000, Priority: 64
    Hello interval: 10 seconds, Hello multiplier: 3, Hold time: 30 seconds
    Designated IS: isr2.01(1980.5110.0121.01)
    Adjacencies: 1
    CNSP in: 2, CNSP out: 0, CNSP success: 2, CNSP fail: 0
    PSNP in: 0, PSNP out: 0, PSNP success: 0, PSNP fail: 0
    LSP in: 0, LSP out: 0, LSP success: 0, LSP fail: 0, LSP in purge: 0 LSP in auth
fail: 0
    IIH in: 12, IIH out: 12
  Interface: ifl-0/0/2/100, Level: 2
    Type: broadcast, Passive: False, Unnumbered: False, Circuit ID: 0x1
    Metric: 1000000, Priority: 64
    Hello interval: 10 seconds, Hello multiplier: 3, Hold time: 30 seconds
    Designated IS: isr2.01(1980.5110.0121.01)
    Adjacencies: 1
    CNSP in: 2, CNSP out: 0, CNSP success: 2, CNSP fail: 0
    PSNP in: 0, PSNP out: 0, PSNP success: 0, PSNP fail: 0
    LSP in: 0, LSP out: 0, LSP success: 0, LSP fail: 0, LSP in purge: 0 LSP in auth
fail: 0
    IIH in: 12, IIH out: 12
```

Example 6: Detailed output of an IS-IS interface that supports broadcast

```
supervisor@rtbrick: op> show isis interface ifl-0/0/2/0
Instance: default
  Interface: ifl-0/0/2/0, Level: 1
    Type: point-to-point, Passive: False, Unnumbered: False
    Metric: 1000000
    Hello interval: 10 seconds, Hello multiplier: 3, Hold time: 30 seconds
    Adjacencies: 1
    CNSP in: 2, CNSP out: 3, CNSP success: 2, CNSP fail: 0
    PSNP in: 1, PSNP out: 1, PSNP success: 1, PSNP fail: 0
    LSP in: 2, LSP out: 0, LSP success: 2, LSP fail: 0, LSP in purge: 0 LSP in auth
fail: 0
    IIH in: 4, IIH out: 5
  Interface: ifl-0/0/2/0, Level: 2
    Type: point-to-point, Passive: False, Unnumbered: False
    Metric: 1000000
    Hello interval: 10 seconds, Hello multiplier: 3, Hold time: 30 seconds
    Adjacencies: 1
    CNSP in: 2, CNSP out: 3, CNSP success: 2, CNSP fail: 0
    PSNP in: 3, PSNP out: 3, PSNP success: 3, PSNP fail: 0
    LSP in: 4, LSP out: 0, LSP success: 4, LSP fail: 0, LSP in purge: 0 LSP in auth
fail: 0
```

```
    IIH in: 4, IIH out: 5
```

API To access the Operational State API that corresponds to this CLI, click include::common:partial$operational_api_links.adoc[tag=show_isis_interface].

#### IS-IS Neighbor

**Syntax:**

**show isis neighbor** <option>

| Option | Description |
|---|---|
| - | Without any option, this command displays a summary of all the IS-IS neighbors |
| detail | Displays detailed information for IS-IS neighbor |
| instance | Displays IS-IS neighbor information for an instance |

Example 1: Summary view of the IS-IS neighbor

```
supervisor@rtbrick: op> show isis neighbor
Instance: default
  Interface          System                   Level   State   Type       Up since
Expires
  ifl-0/0/2/0        isr2                     L1L2    Up      P2P        Fri Jun 05
14:46:38    in 26s 282325us
  ifl-0/0/2/100      isr2                     L1      Up      Broadcast  Fri Jun 05
14:46:38    in 26s 81964us
  ifl-0/0/2/100      isr2                     L2      Up      Broadcast  Fri Jun 05
14:46:40    in 26s 81794us
```

Example 2: Detailed view of the IS-IS neighbor

```
supervisor@rtbrick: op> show isis neighbor detail
Instance: default
  System: isr2, Interface: ifl-0/0/2/0
    State: Up, Level: L1L2, Adjacency type: P2P
    Circuit type: L1L2, Hold time: 30.0s, Expiry time: in 25s 207558us
    Local IPv4 address: 192.0.2.1, Remote IPv4 address: 192.0.2.2, Unnumbered: False
    Local IPv6 address: fe80::9819:2cff:fe06:b503, Remote IPv6 address:
fe80::9819:2cff:fe99:472d
    IPv4 Adjacency SID: 20112, IPv6 Adjacency SID: 20312
    Up since: Fri Jun 05 14:46:38 GMT +0000 2026, Last down reason: NA
    Last transition: 2026-06-05T14:46:38.990940+0000, Number of transitions: 2
    Error counters:
      Level mismatch: 0, Area mismatch: 0, System ID: 0, Subnet mismatch: 0
      Hold timeout: 0, Neighbor down: 0, Interface down: 0, Admin reset: 0, BFD down:
```

```
0
      Interface configuration: 0, Area configuration: 0, Other: 0
  System: isr2, Interface: ifl-0/0/2/100
    State: Up, Level: L1, Adjacency type: Broadcast
    Circuit type: L1L2, Hold time: 30.0s, Expiry time: in 25s 7151us
    Priority: 64, Designated IS: isr2.01(1980.5110.0121.01), SNPA: 98:19:2c:99:47:2d
    Local IPv4 address: 192.0.2.5, Remote IPv4 address: 192.0.2.6, Unnumbered: False
    Local IPv6 address: fe80::9819:2c00:6406:b503, Remote IPv6 address:
fe80::9819:2c00:6499:472d
    Up since: Fri Jun 05 14:46:38 GMT +0000 2026, Last down reason: NA
    Last transition: 2026-06-05T14:46:39.518114+0000, Number of transitions: 3
    Error counters:
      Level mismatch: 0, Area mismatch: 0, System ID: 0, Subnet mismatch: 0
      Hold timeout: 0, Neighbor down: 0, Interface down: 0, Admin reset: 0, BFD down:
0
      Interface configuration: 0, Area configuration: 0, Other: 0
  System: isr2, Interface: ifl-0/0/2/100
    State: Up, Level: L2, Adjacency type: Broadcast
    Circuit type: L1L2, Hold time: 30.0s, Expiry time: in 25s 6941us
    Priority: 64, Designated IS: isr2.01(1980.5110.0121.01), SNPA: 98:19:2c:99:47:2d
    Local IPv4 address: 192.0.2.5, Remote IPv4 address: 192.0.2.6, Unnumbered: False
    Local IPv6 address: fe80::9819:2c00:6406:b503, Remote IPv6 address:
fe80::9819:2c00:6499:472d
    Up since: Fri Jun 05 14:46:40 GMT +0000 2026, Last down reason: NA
    Last transition: 2026-06-05T14:46:40.764532+0000, Number of transitions: 3
    Error counters:
      Level mismatch: 0, Area mismatch: 0, System ID: 0, Subnet mismatch: 0
      Hold timeout: 0, Neighbor down: 0, Interface down: 0, Admin reset: 0, BFD down:
0
      Interface configuration: 0, Area configuration: 0, Other: 0
```

Example 3: Summary view of the IS-IS neighbor for the specified instance

```
supervisor@rtbrick: op> show isis neighbor instance default
Instance: default
  Interface          System                   Level   State   Type       Up since
Expires
  ifl-0/0/2/0        isr2                     L1L2    Up      P2P        Fri Jun 05
14:46:38    in 24s 182062us
  ifl-0/0/2/100      isr2                     L1      Up      Broadcast  Fri Jun 05
14:46:38    in 23s 981695us
  ifl-0/0/2/100      isr2                     L2      Up      Broadcast  Fri Jun 05
14:46:40    in 23s 981520us
```

Example 4: Detailed view of the IS-IS neighbor for the specified instance

```
supervisor@rtbrick: op> show isis neighbor instance default detail
Instance: default
  System: isr2, Interface: ifl-0/0/2/0
```

```
    State: Up, Level: L1L2, Adjacency type: P2P
    Circuit type: L1L2, Hold time: 30.0s, Expiry time: in 23s 173876us
    Local IPv4 address: 192.0.2.1, Remote IPv4 address: 192.0.2.2, Unnumbered: False
    Local IPv6 address: fe80::9819:2cff:fe06:b503, Remote IPv6 address:
fe80::9819:2cff:fe99:472d
    IPv4 Adjacency SID: 20112, IPv6 Adjacency SID: 20312
    Up since: Fri Jun 05 14:46:38 GMT +0000 2026, Last down reason: NA
    Last transition: 2026-06-05T14:46:38.990940+0000, Number of transitions: 2
    Error counters:
      Level mismatch: 0, Area mismatch: 0, System ID: 0, Subnet mismatch: 0
      Hold timeout: 0, Neighbor down: 0, Interface down: 0, Admin reset: 0, BFD down:
0
      Interface configuration: 0, Area configuration: 0, Other: 0
  System: isr2, Interface: ifl-0/0/2/100
    State: Up, Level: L1, Adjacency type: Broadcast
    Circuit type: L1L2, Hold time: 30.0s, Expiry time: in 22s 973463us
    Priority: 64, Designated IS: isr2.01(1980.5110.0121.01), SNPA: 98:19:2c:99:47:2d
    Local IPv4 address: 192.0.2.5, Remote IPv4 address: 192.0.2.6, Unnumbered: False
    Local IPv6 address: fe80::9819:2c00:6406:b503, Remote IPv6 address:
fe80::9819:2c00:6499:472d
    Up since: Fri Jun 05 14:46:38 GMT +0000 2026, Last down reason: NA
    Last transition: 2026-06-05T14:46:39.518114+0000, Number of transitions: 3
    Error counters:
      Level mismatch: 0, Area mismatch: 0, System ID: 0, Subnet mismatch: 0
      Hold timeout: 0, Neighbor down: 0, Interface down: 0, Admin reset: 0, BFD down:
0
      Interface configuration: 0, Area configuration: 0, Other: 0
  System: isr2, Interface: ifl-0/0/2/100
    State: Up, Level: L2, Adjacency type: Broadcast
    Circuit type: L1L2, Hold time: 30.0s, Expiry time: in 22s 973251us
    Priority: 64, Designated IS: isr2.01(1980.5110.0121.01), SNPA: 98:19:2c:99:47:2d
    Local IPv4 address: 192.0.2.5, Remote IPv4 address: 192.0.2.6, Unnumbered: False
    Local IPv6 address: fe80::9819:2c00:6406:b503, Remote IPv6 address:
fe80::9819:2c00:6499:472d
    Up since: Fri Jun 05 14:46:40 GMT +0000 2026, Last down reason: NA
    Last transition: 2026-06-05T14:46:40.764532+0000, Number of transitions: 3
    Error counters:
      Level mismatch: 0, Area mismatch: 0, System ID: 0, Subnet mismatch: 0
      Hold timeout: 0, Neighbor down: 0, Interface down: 0, Admin reset: 0, BFD down:
0
      Interface configuration: 0, Area configuration: 0, Other: 0
```

API To access the Operational State API that corresponds to this CLI, click include::common:partial$operational_api_links.adoc[tag=show_isis_neighbor].

#### Designated IS (DIS) Election History

In IS-IS broadcast networks, the Designated Intermediate System (DIS) is elected to optimize the distribution of routing information, reducing network overhead by coordinating the flooding of Link State Packets (LSPs) among multiple routers. This ensures efficient and stable network

operation. The following command displays the DIS election history in RBFS.

**Syntax:**

**show isis dis-election history** <attribute> <value>

| Option | Description |
|---|---|
| - | Without any option, this command displays a summary of all the IS-IS. |
| instance | Displays DIS history information for the specified instance. |
| interface <interface-name> | Displays IS-IS DIS history information for the specified interface. |
| level [level-1/level-2] | Displays IS-IS DIS history information for the specified level. |

Example: Summary view of IS-IS DIS Election History

```
supervisor@rtbrick: op> show isis dis-election history
Instance: default, Interface: ifl-0/0/2/100, Level: 1
Timestamp                            DIS                      Event
Fri Jun 05 14:46:38 GMT +0000 2026     -                      Neighbour
1980.5110.0121 adjacency state changed from Init to Up
Fri Jun 05 14:46:39 GMT +0000 2026    1980.5110.0121.01        Neighbour
1980.5110.0121 DIS changed from 0000.0000.0000.00 to 1980.5110.0121.01
Instance: default, Interface: ifl-0/0/2/100, Level: 2
Timestamp                            DIS                      Event
Fri Jun 05 14:46:40 GMT +0000 2026     -                      Neighbour
1980.5110.0121 adjacency state changed from Init to Up
Fri Jun 05 14:46:40 GMT +0000 2026    1980.5110.0121.01        Neighbour
1980.5110.0121 DIS changed from 0000.0000.0000.00 to 1980.5110.0121.01
```

#### IS-IS Hostname

**Syntax:**

**show isis hostname**

| Option | Description |
|---|---|
| - | Without any option, this command displays a summary of all the IS-IS dynamic hostnames |

Example: Summary view of IS-IS hostnames

```
supervisor@rtbrick: op> show isis hostname
Instance       System-ID           Hostname
```

```
default        1980.5110.0111      isr1
default        1980.5110.0121      isr2
```

#### IS-IS Database

**Syntax:**

**show isis database** <option>

| Option | Description |
|---|---|
| - | Without any option, this command displays all the IS-IS databases |
| detail | Displays detailed information for IS-IS database |
| instance | Displays IS-IS database information for an instance |
| lsp <lsp_name> | Displays a summary of IS-IS database for the specified LSP name. |
| lsp id <lsp-id> | Displays a summary of IS-IS database for the specified LSP ID. |
| [level-1/level-2] lsp | Displays a summary of IS-IS database LSP information for specified level |
| system <system-id> | Displays a summary of IS-IS database for all LSPs from a system |
| [level-1/level-2] system | Displays a summary of IS-IS database for all LSPs from a system on the specified level. |
| [level-1/level-2] detail | Displays detailed information for the specified level |

Example 1: Summary view of the IS-IS Database

```
supervisor@rtbrick: op> show isis database
Instance: default, Level: 1
  LSP ID                    Sequence     Checksum     Lifetime     Overload
Attached
  isr1.00-00                     0x4       0x8517        65514            0
0
  isr2.00-00                     0x3        0x1c4        65514            0
0
  isr2.01-00                     0x1       0x2cff        65514            0
0
Instance: default, Level: 2
  LSP ID                    Sequence     Checksum     Lifetime     Overload
Attached
  isr1.00-00                     0x6       0x7696        65514            0
0
  isr2.00-00                     0x6       0x44fc        65514            0
0
  isr2.01-00                     0x1       0x8e6f        65514            0
```

```
0
```

Example 2: Summary view of the IS-IS database for the specified LSP

```
supervisor@rtbrick: op> show isis database lsp id 1980.5110.0111.00-00
Instance: default, Level: 1
  LSP ID: isr1.00-00
    Interface:
    LSP Header:
      Sequence: 0x4
      Checksum: 0x8517
      Remaining lifetime: 65535 seconds
      Flags: Attached: 0, Overload: 0
    Packet:
      Length: 248 bytes
      Last received time: 2026-06-05T14:46:39.731129+0000
      Expiry: expires in 18h 11m 52s 625218us
    System ID: 1980.5110.0111
    Dynamic Hostname TLV (137): isr1
    Protocols Supported TLVs (129):
      Network layer protocol ID: IPv6
      Network layer protocol ID: IPv4
    Area Address TLVs (1):
      Area address: 49.0001
    Authentication TLV (10):
      Type: md5
      Value: 65dfacffdbdbfa061d402f51d41b358d
    IS Reachability TLVs (22):
      IS neighbor: 1980.5110.0121.00              Metric:  1000000
        Adjacency SID:
          Value:      20112, Flags: Value, Local, Persistent
          Value:      20312, Flags: Ipv6 Encapsulation, Value, Local, Persistent
      IS neighbor: 1980.5110.0121.01              Metric:  1000000
    IPv4 Reachability TLVs (135):
      IPv4 prefix: 192.0.2.0/30                   Metric:  1000000   Internal   Up
      IPv4 prefix: 192.0.2.4/30                   Metric:  1000000   Internal   Up
      IPv4 prefix: 198.51.100.11/32               Metric:  1000000   Internal   Up
SID:  100   Flags: Node
      IPv4 prefix: 198.51.100.12/32               Metric:  1000000   Internal   Up
    IPv6 Reachability TLVs (236):
      IPv6 prefix: 2001:db8:1:1::/64              Metric:  1000000   Internal   Up
      IPv6 prefix: 2001:db8:2:1::/64              Metric:  1000000   Internal   Up
      IPv6 prefix: 2001:db8:0:1::11/128           Metric:  1000000   Internal   Up
SID:  101   Flags: Node
      IPv6 prefix: 2001:db8:0:1::12/128           Metric:  1000000   Internal   Up
    Segment Routing TLVs (242/sub 2):
      SRGB: Base: 10000, Range: 1000, Router ID: 198.51.100.11
Instance: default, Level: 2
  LSP ID: isr1.00-00
    Interface:
```

```
    LSP Header:
      Sequence: 0x6
      Checksum: 0x7696
      Remaining lifetime: 65535 seconds
      Flags: Attached: 0, Overload: 0
    Packet:
      Length: 342 bytes
      Last received time: 2026-06-05T14:46:40.976088+0000
      Expiry: expires in 18h 11m 53s 824945us
    System ID: 1980.5110.0111
    Dynamic Hostname TLV (137): isr1
    Protocols Supported TLVs (129):
      Network layer protocol ID: IPv6
      Network layer protocol ID: IPv4
    Area Address TLVs (1):
      Area address: 49.0001
    Authentication TLV (10):
      Type: md5
      Value: 5a9b30d2500f2c55f71c8b3636e3c631
    IS Reachability TLVs (22):
      IS neighbor: 1980.5110.0121.00              Metric:  1000000
        Adjacency SID:
          Value:      20112, Flags: Value, Local, Persistent
          Value:      20312, Flags: Ipv6 Encapsulation, Value, Local, Persistent
      IS neighbor: 1980.5110.0121.01              Metric:  1000000
    IPv4 Reachability TLVs (135):
      IPv4 prefix: 192.0.2.0/30                   Metric:  1000000   Internal   Up
      IPv4 prefix: 192.0.2.4/30                   Metric:  1000000   Internal   Up
      IPv4 prefix: 198.51.100.11/32               Metric:  1000000   Internal   Up
SID:  100   Flags: Node
      IPv4 prefix: 198.51.100.12/32               Metric:  1000000   Internal   Up
      IPv4 prefix: 198.51.100.21/32               Metric:  2000000   Internal   Up
SID:  200   Flags: Re-advertisement, Node
      IPv4 prefix: 198.51.100.22/32               Metric:  2000000   Internal   Up
    IPv6 Reachability TLVs (236):
      IPv6 prefix: 2001:db8:1:1::/64              Metric:  1000000   Internal   Up
      IPv6 prefix: 2001:db8:2:1::/64              Metric:  1000000   Internal   Up
      IPv6 prefix: 2001:db8:0:1::11/128           Metric:  1000000   Internal   Up
SID:  101   Flags: Node
      IPv6 prefix: 2001:db8:0:1::12/128           Metric:  1000000   Internal   Up
      IPv6 prefix: 2001:db8:0:1::21/128           Metric:  2000000   Internal   Up
SID:  201   Flags: Re-advertisement, Node
      IPv6 prefix: 2001:db8:0:1::22/128           Metric:  2000000   Internal   Up
    Segment Routing TLVs (242/sub 2):
      SRGB: Base: 10000, Range: 1000, Router ID: 198.51.100.11
```

Example 3: Detailed view of the IS-IS database for level-1

```
supervisor@rtbrick: op> show isis database level-1 detail
Instance: default, Level: 1
```

```
  LSP ID: isr1.00-00
    Interface:
    LSP Header:
      Sequence: 0x4
      Checksum: 0x8517
      Remaining lifetime: 65511 seconds
      Flags: Attached: 0, Overload: 0
    Packet:
      Length: 248 bytes
      Last received time: 2026-06-05T14:46:39.731129+0000
      Expiry: expires in 18h 11m 51s 541232us
    System ID: 1980.5110.0111
    Dynamic Hostname TLV (137): isr1
    Protocols Supported TLVs (129):
      Network layer protocol ID: IPv6
      Network layer protocol ID: IPv4
    Area Address TLVs (1):
      Area address: 49.0001
    Authentication TLV (10):
      Type: md5
      Value: 65dfacffdbdbfa061d402f51d41b358d
    IS Reachability TLVs (22):
      IS neighbor: 1980.5110.0121.00              Metric:  1000000
        Adjacency SID:
          Value:      20112, Flags: Value, Local, Persistent
          Value:      20312, Flags: Ipv6 Encapsulation, Value, Local, Persistent
      IS neighbor: 1980.5110.0121.01              Metric:  1000000
    IPv4 Reachability TLVs (135):
      IPv4 prefix: 192.0.2.0/30                   Metric:  1000000   Internal   Up
      IPv4 prefix: 192.0.2.4/30                   Metric:  1000000   Internal   Up
      IPv4 prefix: 198.51.100.11/32               Metric:  1000000   Internal   Up
SID:  100   Flags: Node
      IPv4 prefix: 198.51.100.12/32               Metric:  1000000   Internal   Up
    IPv6 Reachability TLVs (236):
      IPv6 prefix: 2001:db8:1:1::/64              Metric:  1000000   Internal   Up
      IPv6 prefix: 2001:db8:2:1::/64              Metric:  1000000   Internal   Up
      IPv6 prefix: 2001:db8:0:1::11/128           Metric:  1000000   Internal   Up
SID:  101   Flags: Node
      IPv6 prefix: 2001:db8:0:1::12/128           Metric:  1000000   Internal   Up
    Segment Routing TLVs (242/sub 2):
      SRGB: Base: 10000, Range: 1000, Router ID: 198.51.100.11
  LSP ID: isr2.00-00
    Interface: ifl-0/0/2/0
    LSP Header:
      Sequence: 0x3
      Checksum: 0x1c4
      Remaining lifetime: 65511 seconds
      Flags: Attached: 0, Overload: 0
    Packet:
      Length: 248 bytes
      Last received time: 2026-06-05T14:46:39.411068+0000
```

```
      Expiry: expires in 18h 11m 51s 220843us
    System ID: 1980.5110.0121
    Dynamic Hostname TLV (137): isr2
    Protocols Supported TLVs (129):
      Network layer protocol ID: IPv6
      Network layer protocol ID: IPv4
    Area Address TLVs (1):
      Area address: 49.0001
    Authentication TLV (10):
      Type: md5
      Value: 1f82cbc53859f87066a51805990662d7
    IS Reachability TLVs (22):
      IS neighbor: 1980.5110.0111.00              Metric:  1000000
        Adjacency SID:
          Value:      21121, Flags: Value, Local, Persistent
          Value:      21321, Flags: Ipv6 Encapsulation, Value, Local, Persistent
      IS neighbor: 1980.5110.0121.01              Metric:  1000000
    IPv4 Reachability TLVs (135):
      IPv4 prefix: 192.0.2.0/30                   Metric:  1000000   Internal   Up
      IPv4 prefix: 192.0.2.4/30                   Metric:  1000000   Internal   Up
      IPv4 prefix: 198.51.100.21/32               Metric:  1000000   Internal   Up
SID:  200   Flags: Node
      IPv4 prefix: 198.51.100.22/32               Metric:  1000000   Internal   Up
    IPv6 Reachability TLVs (236):
      IPv6 prefix: 2001:db8:1:1::/64              Metric:  1000000   Internal   Up
      IPv6 prefix: 2001:db8:2:1::/64              Metric:  1000000   Internal   Up
      IPv6 prefix: 2001:db8:0:1::21/128           Metric:  1000000   Internal   Up
SID:  201   Flags: Node
      IPv6 prefix: 2001:db8:0:1::22/128           Metric:  1000000   Internal   Up
    Segment Routing TLVs (242/sub 2):
      SRGB: Base: 11000, Range: 1000, Router ID: 198.51.100.21
  LSP ID: isr2.01-00
    Interface: ifl-0/0/2/0
    LSP Header:
      Sequence: 0x1
      Checksum: 0x2cff
      Remaining lifetime: 65511 seconds
      Flags: Attached: 0, Overload: 0
    Packet:
      Length: 80 bytes
      Last received time: 2026-06-05T14:46:39.411333+0000
      Expiry: expires in 18h 11m 51s 220905us
    System ID: 1980.5110.0121
    Protocols Supported TLVs (129):
      Network layer protocol ID: IPv6
      Network layer protocol ID: IPv4
    Area Address TLVs (1):
      Area address: 49.0001
    Authentication TLV (10):
      Type: md5
      Value: d6d4ece4874d0946847e3e9a81621364
```

```
    IS Reachability TLVs (22):
      IS neighbor: 1980.5110.0111.00              Metric:        0
      IS neighbor: 1980.5110.0121.00              Metric:        0
```

Example 4: Summary view of the IS-IS database for the specified instance

```
supervisor@rtbrick: op> show isis database instance default
Instance: default, Level: 1
  LSP ID                    Sequence     Checksum     Lifetime     Overload
Attached
  isr1.00-00                     0x4       0x8517        65510            0
0
  isr2.00-00                     0x3        0x1c4        65510            0
0
  isr2.01-00                     0x1       0x2cff        65510            0
0
Instance: default, Level: 2
  LSP ID                    Sequence     Checksum     Lifetime     Overload
Attached
  isr1.00-00                     0x6       0x7696        65511            0
0
  isr2.00-00                     0x6       0x44fc        65511            0
0
  isr2.01-00                     0x1       0x8e6f        65511            0
0
```

Example 5: Summary view of the IS-IS database for the specified instance and LSP

```
supervisor@rtbrick: op> show isis database instance default lsp id 1980.5110.0111.00-
00
Instance: default, Level: 1
  LSP ID: isr1.00-00
    Interface:
    LSP Header:
      Sequence: 0x4
      Checksum: 0x8517
      Remaining lifetime: 65535 seconds
      Flags: Attached: 0, Overload: 0
    Packet:
      Length: 248 bytes
      Last received time: 2026-06-05T14:46:39.731129+0000
      Expiry: expires in 18h 11m 49s 461377us
    System ID: 1980.5110.0111
    Dynamic Hostname TLV (137): isr1
    Protocols Supported TLVs (129):
      Network layer protocol ID: IPv6
      Network layer protocol ID: IPv4
    Area Address TLVs (1):
      Area address: 49.0001
```

```
    Authentication TLV (10):
      Type: md5
      Value: 65dfacffdbdbfa061d402f51d41b358d
    IS Reachability TLVs (22):
      IS neighbor: 1980.5110.0121.00              Metric:  1000000
        Adjacency SID:
          Value:      20112, Flags: Value, Local, Persistent
          Value:      20312, Flags: Ipv6 Encapsulation, Value, Local, Persistent
      IS neighbor: 1980.5110.0121.01              Metric:  1000000
    IPv4 Reachability TLVs (135):
      IPv4 prefix: 192.0.2.0/30                   Metric:  1000000   Internal   Up
      IPv4 prefix: 192.0.2.4/30                   Metric:  1000000   Internal   Up
      IPv4 prefix: 198.51.100.11/32               Metric:  1000000   Internal   Up
SID:  100   Flags: Node
      IPv4 prefix: 198.51.100.12/32               Metric:  1000000   Internal   Up
    IPv6 Reachability TLVs (236):
      IPv6 prefix: 2001:db8:1:1::/64              Metric:  1000000   Internal   Up
      IPv6 prefix: 2001:db8:2:1::/64              Metric:  1000000   Internal   Up
      IPv6 prefix: 2001:db8:0:1::11/128           Metric:  1000000   Internal   Up
SID:  101   Flags: Node
      IPv6 prefix: 2001:db8:0:1::12/128           Metric:  1000000   Internal   Up
    Segment Routing TLVs (242/sub 2):
      SRGB: Base: 10000, Range: 1000, Router ID: 198.51.100.11
Instance: default, Level: 2
  LSP ID: isr1.00-00
    Interface:
    LSP Header:
      Sequence: 0x6
      Checksum: 0x7696
      Remaining lifetime: 65535 seconds
      Flags: Attached: 0, Overload: 0
    Packet:
      Length: 342 bytes
      Last received time: 2026-06-05T14:46:40.976088+0000
      Expiry: expires in 18h 11m 50s 661866us
    System ID: 1980.5110.0111
    Dynamic Hostname TLV (137): isr1
    Protocols Supported TLVs (129):
      Network layer protocol ID: IPv6
      Network layer protocol ID: IPv4
    Area Address TLVs (1):
      Area address: 49.0001
    Authentication TLV (10):
      Type: md5
      Value: 5a9b30d2500f2c55f71c8b3636e3c631
    IS Reachability TLVs (22):
      IS neighbor: 1980.5110.0121.00              Metric:  1000000
        Adjacency SID:
          Value:      20112, Flags: Value, Local, Persistent
          Value:      20312, Flags: Ipv6 Encapsulation, Value, Local, Persistent
      IS neighbor: 1980.5110.0121.01              Metric:  1000000
```

```
    IPv4 Reachability TLVs (135):
      IPv4 prefix: 192.0.2.0/30                  Metric:  1000000   Internal   Up
      IPv4 prefix: 192.0.2.4/30                  Metric:  1000000   Internal   Up
      IPv4 prefix: 198.51.100.11/32              Metric:  1000000   Internal   Up
SID:  100   Flags: Node
      IPv4 prefix: 198.51.100.12/32              Metric:  1000000   Internal   Up
      IPv4 prefix: 198.51.100.21/32              Metric:  2000000   Internal   Up
SID:  200   Flags: Re-advertisement, Node
      IPv4 prefix: 198.51.100.22/32              Metric:  2000000   Internal   Up
    IPv6 Reachability TLVs (236):
      IPv6 prefix: 2001:db8:1:1::/64             Metric:  1000000   Internal   Up
      IPv6 prefix: 2001:db8:2:1::/64             Metric:  1000000   Internal   Up
      IPv6 prefix: 2001:db8:0:1::11/128          Metric:  1000000   Internal   Up
SID:  101   Flags: Node
      IPv6 prefix: 2001:db8:0:1::12/128          Metric:  1000000   Internal   Up
      IPv6 prefix: 2001:db8:0:1::21/128          Metric:  2000000   Internal   Up
SID:  201   Flags: Re-advertisement, Node
      IPv6 prefix: 2001:db8:0:1::22/128          Metric:  2000000   Internal   Up
    Segment Routing TLVs (242/sub 2):
      SRGB: Base: 10000, Range: 1000, Router ID: 198.51.100.11
```

#### IS-IS Route

**Syntax:**

**show isis route** <option>

| Option | Description |
|---|---|
| - | Without any option, this command displays a summary of the IS-IS route information |
| instance | Displays IS-IS route information for an instance |
| <afi> <safi> | Routing information for the specified AFI/SAFI. Supported SAFI values are 'unicast' and 'labeled-unicast'. |

Example 1: Summary view of the IS-IS routes

```
supervisor@rtbrick: op> show isis route
Instance: default, AFI: ipv4, SAFI: unicast
  Prefix                    Level   Metric     Type         Next Hop
Interface
  192.0.2.0/30                  1   1000000    Internal     n/a
local
  192.0.2.4/30                  1   1000000    Internal     n/a
local
  198.51.100.11/32              1   1000000    Internal     n/a
local
  198.51.100.12/32              1   1000000    Internal     n/a
```

```
local
  198.51.100.21/32              1   2000000    Internal     192.0.2.2
ifl-0/0/2/0
                                                            192.0.2.6
ifl-0/0/2/100
  198.51.100.22/32              1   2000000    Internal     192.0.2.2
ifl-0/0/2/0
                                                            192.0.2.6
ifl-0/0/2/100
Instance: default, AFI: ipv4, SAFI: labeled-unicast
  Prefix                    Level   Metric     SID Index    Next Hop
Interface            Label
  198.51.100.11/32              1   1000000    100          n/a
local                10100
  198.51.100.21/32              1   2000000    200          192.0.2.2
ifl-0/0/2/0          10200
                                                            192.0.2.6
ifl-0/0/2/100        10200
Instance: default, AFI: ipv6, SAFI: unicast
  Prefix                    Level   Metric     Type         Next Hop
Interface
  2001:db8:0:1::11/128          1   1000000    Internal     n/a
local
  2001:db8:0:1::12/128          1   1000000    Internal     n/a
local
  2001:db8:0:1::21/128          1   2000000    Internal     fe80::9819:2cff:fe99:472d
ifl-0/0/2/0
                                                            fe80::9819:2c00:6499:472d
ifl-0/0/2/100
  2001:db8:0:1::22/128          1   2000000    Internal     fe80::9819:2cff:fe99:472d
ifl-0/0/2/0
                                                            fe80::9819:2c00:6499:472d
ifl-0/0/2/100
  2001:db8:1:1::/64             1   1000000    Internal     n/a
local
  2001:db8:2:1::/64             1   1000000    Internal     n/a
local
Instance: default, AFI: ipv6, SAFI: labeled-unicast
  Prefix                    Level   Metric     SID Index    Next Hop
Interface            Label
  2001:db8:0:1::11/128          1   1000000    101          n/a
local                10101
  2001:db8:0:1::21/128          1   2000000    201          fe80::9819:2cff:fe99:472d
ifl-0/0/2/0          10201
                                                            fe80::9819:2c00:6499:472d
ifl-0/0/2/100        10201
Instance: default, AFI: mpls
  Label                     Level   Next Hop                  Interface
  label:10100                   1   n/a                       local
  label:10100,bos:1             1   n/a                       local
  label:10101                   1   n/a                       local
```

```
  label:10101,bos:1             1   n/a                       local
  label:10200                   1   192.0.2.2                 ifl-0/0/2/0
                                    192.0.2.6                 ifl-0/0/2/100
  label:10200,bos:1             1   192.0.2.2                 ifl-0/0/2/0
                                    192.0.2.6                 ifl-0/0/2/100
  label:10201                   1   fe80::9819:2cff:fe99:472d ifl-0/0/2/0
                                    fe80::9819:2c00:6499:472d ifl-0/0/2/100
  label:10201,bos:1             1   fe80::9819:2cff:fe99:472d ifl-0/0/2/0
                                    fe80::9819:2c00:6499:472d ifl-0/0/2/100
  label:20112                   1   192.0.2.2                 ifl-0/0/2/0
  label:20112,bos:1             1   192.0.2.2                 ifl-0/0/2/0
  label:20312                   1   fe80::9819:2cff:fe99:472d ifl-0/0/2/0
  label:20312,bos:1             1   fe80::9819:2cff:fe99:472d ifl-0/0/2/0
```

Example 2: Summary view of the IS-IS routes for the specified instance

```
supervisor@rtbrick: op> show isis route instance default
Instance: default, AFI: ipv4, SAFI: unicast
  Prefix                    Level   Metric     Type         Next Hop
Interface
  192.0.2.0/30                  1   1000000    Internal     n/a
local
  192.0.2.4/30                  1   1000000    Internal     n/a
local
  198.51.100.11/32              1   1000000    Internal     n/a
local
  198.51.100.12/32              1   1000000    Internal     n/a
local
  198.51.100.21/32              1   2000000    Internal     192.0.2.2
ifl-0/0/2/0
                                                             192.0.2.6
ifl-0/0/2/100
  198.51.100.22/32              1   2000000    Internal     192.0.2.2
ifl-0/0/2/0
                                                             192.0.2.6
ifl-0/0/2/100
Instance: default, AFI: ipv4, SAFI: labeled-unicast
  Prefix                    Level   Metric     SID Index    Next Hop
Interface            Label
  198.51.100.11/32              1   1000000    100          n/a
local                10100
  198.51.100.21/32              1   2000000    200          192.0.2.2
ifl-0/0/2/0          10200
                                                             192.0.2.6
ifl-0/0/2/100        10200
Instance: default, AFI: ipv6, SAFI: unicast
  Prefix                    Level   Metric     Type         Next Hop
Interface
  2001:db8:0:1::11/128          1   1000000    Internal     n/a
local
```

```
  2001:db8:0:1::12/128          1   1000000    Internal     n/a
local
  2001:db8:0:1::21/128          1   2000000    Internal     fe80::9819:2cff:fe99:472d
ifl-0/0/2/0
                                                            fe80::9819:2c00:6499:472d
ifl-0/0/2/100
  2001:db8:0:1::22/128          1   2000000    Internal     fe80::9819:2cff:fe99:472d
ifl-0/0/2/0
                                                            fe80::9819:2c00:6499:472d
ifl-0/0/2/100
  2001:db8:1:1::/64             1   1000000    Internal     n/a
local
  2001:db8:2:1::/64             1   1000000    Internal     n/a
local
Instance: default, AFI: ipv6, SAFI: labeled-unicast
  Prefix                    Level   Metric     SID Index    Next Hop
Interface            Label
  2001:db8:0:1::11/128          1   1000000    101          n/a
local                10101
  2001:db8:0:1::21/128          1   2000000    201          fe80::9819:2cff:fe99:472d
ifl-0/0/2/0          10201
                                                            fe80::9819:2c00:6499:472d
ifl-0/0/2/100        10201
Instance: default, AFI: mpls
  Label                     Level   Next Hop                  Interface
  label:10100                   1   n/a                       local
  label:10100,bos:1             1   n/a                       local
  label:10101                   1   n/a                       local
  label:10101,bos:1             1   n/a                       local
  label:10200                   1   192.0.2.2                 ifl-0/0/2/0
                                    192.0.2.6                 ifl-0/0/2/100
  label:10200,bos:1             1   192.0.2.2                 ifl-0/0/2/0
                                    192.0.2.6                 ifl-0/0/2/100
  label:10201                   1   fe80::9819:2cff:fe99:472d ifl-0/0/2/0
                                    fe80::9819:2c00:6499:472d ifl-0/0/2/100
  label:10201,bos:1             1   fe80::9819:2cff:fe99:472d ifl-0/0/2/0
                                    fe80::9819:2c00:6499:472d ifl-0/0/2/100
  label:20112                   1   192.0.2.2                 ifl-0/0/2/0
  label:20112,bos:1             1   192.0.2.2                 ifl-0/0/2/0
  label:20312                   1   fe80::9819:2cff:fe99:472d ifl-0/0/2/0
  label:20312,bos:1             1   fe80::9819:2cff:fe99:472d ifl-0/0/2/0
```

Example 3: Summary view of the IS-IS routes for the specified instance and address family (IPv4 unicast).

```
supervisor@rtbrick: op> show isis route instance default ipv4 unicast
Instance: default, AFI: ipv4, SAFI: unicast
  Prefix                    Level   Metric     Type         Next Hop
Interface
  192.0.2.0/30                  1   1000000    Internal     n/a
```

```
local
  192.0.2.4/30                  1   1000000    Internal     n/a
local
  198.51.100.11/32              1   1000000    Internal     n/a
local
  198.51.100.12/32              1   1000000    Internal     n/a
local
  198.51.100.21/32              1   2000000    Internal     192.0.2.2
ifl-0/0/2/0
                                                             192.0.2.6
ifl-0/0/2/100
  198.51.100.22/32              1   2000000    Internal     192.0.2.2
ifl-0/0/2/0
                                                             192.0.2.6
ifl-0/0/2/100
```

Example 4: Summary view of the IS-IS routes for the specified instance and address family (IPv6 unicast).

```
supervisor@rtbrick: op> show isis route instance default ipv6 unicast
Instance: default, AFI: ipv6, SAFI: unicast
  Prefix                    Level   Metric     Type         Next Hop
Interface
  2001:db8:0:1::11/128          1   1000000    Internal     n/a
local
  2001:db8:0:1::12/128          1   1000000    Internal     n/a
local
  2001:db8:0:1::21/128          1   2000000    Internal     fe80::9819:2cff:fe99:472d
ifl-0/0/2/0
                                                             fe80::9819:2c00:6499:472d
ifl-0/0/2/100
  2001:db8:0:1::22/128          1   2000000    Internal     fe80::9819:2cff:fe99:472d
ifl-0/0/2/0
                                                             fe80::9819:2c00:6499:472d
ifl-0/0/2/100
  2001:db8:1:1::/64             1   1000000    Internal     n/a
local
  2001:db8:2:1::/64             1   1000000    Internal     n/a
local
```

Example 5: Summary view of the IS-IS routes for the specified instance and address family (IPv4 labeled-unicast).

```
supervisor@rtbrick: op> show isis route instance default ipv4 labeled-unicast
Instance: default, AFI: ipv4, SAFI: labeled-unicast
  Prefix                    Level   Metric     SID Index    Next Hop
Interface            Label
  198.51.100.11/32              1   1000000    100          n/a
```

```
local                10100
  198.51.100.21/32              1   2000000    200          192.0.2.2
ifl-0/0/2/0          10200
                                                            192.0.2.6
ifl-0/0/2/100        10200
```

Example 6: Summary view of the IS-IS routes for the specified instance and address family (IPv6 labeled-unicast).

```
supervisor@rtbrick: op> show isis route instance default ipv6 labeled-unicast
Instance: default, AFI: ipv6, SAFI: labeled-unicast
  Prefix                    Level   Metric     SID Index    Next Hop
Interface            Label
  2001:db8:0:1::11/128          1   1000000    101          n/a
local                10101
  2001:db8:0:1::21/128          1   2000000    201          fe80::9819:2cff:fe99:472d
ifl-0/0/2/0          10201
                                                            fe80::9819:2c00:6499:472d
ifl-0/0/2/100        10201
```

#### IS-IS Segment Routing

**Syntax:**

**show isis segment-routing** <option>

| Option | Description |
|---|---|
| `global-block` | Displays Segment routing global block (SRGB) information |
| `global-block instance <instance>` | Displays Segment routing global block (SRGB) information for the specified instance |
| `label-binding` | Displays the IS-IS segment routing label bindings information |
| `label-binding instance <instance>` | Displays the IS-IS segment routing label bindings for the specified instance |
| `prefix-segment` | Displays the IS-IS prefix segments information |
| `prefix-segment <instance>` | Displays the IS-IS prefix segments for the specified instance |
| `adjacency-segment` | Displays the neighbor information for each SID, which helps users identify to which adjacency the SID belongs. |

Example 1: Summary view of the IS-IS segment routing global block

```
supervisor@rtbrick: op> show isis segment-routing global-block
Instance: default, Level: 1
  System                    SRGB Base   SRGB Range
```

```
  isr1                          10000         1000
  isr2                          11000         1000
Instance: default, Level: 2
  System                     SRGB Base   SRGB Range
  isr1                          10000         1000
  isr2                          11000         1000
```

Example 2: Summary view of the IS-IS segment routing label binding

```
supervisor@rtbrick: op> show isis segment-routing label-binding
Instance: default, Level: 1
  System                     Prefix                              Range   SID     Flags
  isr1                       198.51.100.11/32                        5     100   None
```

Example 3: Summary view of the IS-IS segment routing prefix segment

```
supervisor@rtbrick: op> show isis segment-routing prefix-segment
Instance: default, Level: 1
  System                     Prefix                                  SID     Flags
  isr1                       198.51.100.11/32                          100   Node
  isr1                       2001:db8:0:1::11/128                      101   Node
  isr2                       198.51.100.21/32                          200   Node
  isr2                       2001:db8:0:1::21/128                      201   Node
Instance: default, Level: 2
  System                     Prefix                                  SID     Flags
  isr1                       198.51.100.11/32                          100   Node
  isr1                       198.51.100.21/32                          200   Re-
advertisement, Node
  isr1                       2001:db8:0:1::11/128                      101   Node
  isr1                       2001:db8:0:1::21/128                      201   Re-
advertisement, Node
  isr2                       198.51.100.11/32                          100   Re-
advertisement, Node
  isr2                       198.51.100.21/32                          200   Node
  isr2                       2001:db8:0:1::11/128                      101   Re-
advertisement, Node
  isr2                       2001:db8:0:1::21/128                      201   Node
```

Example 4: Summary view of the IS-IS segment routing adjacency-segment

```
supervisor@rtbrick: op> show isis segment-routing adjacency-segment
Instance: default, Level: 1
  Local System ID            Remote System ID             Label     Interface
Flags
  isr1                       isr2                         20112     ifl-0/0/2/0
Value, Local, Persistent
  isr1                       isr2                         20312     ifl-0/0/2/0
```

```
Ipv6 Encapsulation, Value, Local, Persistent
  isr2                      isr1                          21121     -
Value, Local, Persistent
  isr2                      isr1                          21321     -
Ipv6 Encapsulation, Value, Local, Persistent
Instance: default, Level: 2
  Local System ID           Remote System ID              Label     Interface
Flags
  isr1                      isr2                          20112     ifl-0/0/2/0
Value, Local, Persistent
  isr1                      isr2                          20312     ifl-0/0/2/0
Ipv6 Encapsulation, Value, Local, Persistent
  isr2                      isr1                          21121     -
Value, Local, Persistent
  isr2                      isr1                          21321     -
Ipv6 Encapsulation, Value, Local, Persistent
```

#### IS-IS SPF

**Syntax:**

**show isis spf** <option>

| Option | Description |
|---|---|
| `result` | Displays a summary of the IS-IS SPF results |
| `result <instance>` | Displays a summary of the IS-IS SPF results for the specified instance |
| `result [level-1/level2]` | Displays a summary of the IS-IS SPF results for the specified level. |

Example 1: Summary view of the IS-IS SPF result

```
supervisor@rtbrick: op> show isis spf result
Instance: default, Level: 1
  Destination Node          Metric       Neighbor Node         Interface
Nexthop   Address
  1980.5110.0111.00               0                             local
  1980.5110.0121.00          1000000     1980.5110.0121.00     ifl-0/0/2/0       IPv6
fe80::9819:2cff:fe99:472d
                                         1980.5110.0121.00     ifl-0/0/2/100     IPv6
fe80::9819:2c00:6499:472d
                                         1980.5110.0121.00     ifl-0/0/2/0       IPv4
192.0.2.2
                                         1980.5110.0121.00     ifl-0/0/2/100     IPv4
192.0.2.6
  1980.5110.0121.01          1000000                           local
Instance: default, Level: 2
```

```
  Destination Node          Metric       Neighbor Node         Interface
Nexthop   Address
  1980.5110.0111.00                0                             local
  1980.5110.0121.00          1000000     1980.5110.0121.00     ifl-0/0/2/0       IPv6
fe80::9819:2cff:fe99:472d
                                         1980.5110.0121.00     ifl-0/0/2/100     IPv6
fe80::9819:2c00:6499:472d
                                         1980.5110.0121.00     ifl-0/0/2/0       IPv4
192.0.2.2
                                         1980.5110.0121.00     ifl-0/0/2/100     IPv4
192.0.2.6
  1980.5110.0121.01          1000000                             local
```

Example 2: Summary view of the IS-IS SPF result for level-1

```
supervisor@rtbrick: op> show isis spf result level-1
Instance: default, Level: 1
  Destination Node          Metric       Neighbor Node         Interface
Nexthop   Address
  1980.5110.0111.00                0                             local
  1980.5110.0121.00          1000000     1980.5110.0121.00     ifl-0/0/2/0       IPv6
fe80::9819:2cff:fe99:472d
                                         1980.5110.0121.00     ifl-0/0/2/100     IPv6
fe80::9819:2c00:6499:472d
                                         1980.5110.0121.00     ifl-0/0/2/0       IPv4
192.0.2.2
                                         1980.5110.0121.00     ifl-0/0/2/100     IPv4
192.0.2.6
  1980.5110.0121.01          1000000                             local
```

Example 3: Summary view of the IS-IS SPF result for level-2

```
supervisor@rtbrick: op> show isis spf result level-2
Instance: default, Level: 2
  Destination Node          Metric       Neighbor Node         Interface
Nexthop   Address
  1980.5110.0111.00                0                             local
  1980.5110.0121.00          1000000     1980.5110.0121.00     ifl-0/0/2/0       IPv6
fe80::9819:2cff:fe99:472d
                                         1980.5110.0121.00     ifl-0/0/2/100     IPv6
fe80::9819:2c00:6499:472d
                                         1980.5110.0121.00     ifl-0/0/2/0       IPv4
192.0.2.2
                                         1980.5110.0121.00     ifl-0/0/2/100     IPv4
192.0.2.6
  1980.5110.0121.01          1000000                             local
```

Example 4: Summary view of the IS-IS SPF result of a specific instance for level-1

```
supervisor@rtbrick: op> show isis spf result instance default level-1
Instance: default, Level: 1
  Destination Node          Metric       Neighbor Node         Interface
Nexthop   Address
  1980.5110.0111.00               0                             local
  1980.5110.0121.00          1000000     1980.5110.0121.00     ifl-0/0/2/0       IPv6
fe80::9819:2cff:fe99:472d
                                         1980.5110.0121.00     ifl-0/0/2/100     IPv6
fe80::9819:2c00:6499:472d
                                         1980.5110.0121.00     ifl-0/0/2/0       IPv4
192.0.2.2
                                         1980.5110.0121.00     ifl-0/0/2/100     IPv4
192.0.2.6
  1980.5110.0121.01          1000000                             local
```

Example 5: Summary view of the IS-IS SPF result of a specific instance for level-2

```
supervisor@rtbrick: op> show isis spf result instance default level-2
Instance: default, Level: 2
  Destination Node          Metric       Neighbor Node         Interface
Nexthop   Address
  1980.5110.0111.00               0                             local
  1980.5110.0121.00          1000000     1980.5110.0121.00     ifl-0/0/2/0       IPv6
fe80::9819:2cff:fe99:472d
                                         1980.5110.0121.00     ifl-0/0/2/100     IPv6
fe80::9819:2c00:6499:472d
                                         1980.5110.0121.00     ifl-0/0/2/0       IPv4
192.0.2.2
                                         1980.5110.0121.00     ifl-0/0/2/100     IPv4
192.0.2.6
  1980.5110.0121.01          1000000                             local
```

### IS-IS Clear Commands

Clear commands allow to reset operational states.

#### IS-IS Interface

**Syntax:**

**clear isis interface** <option> ...

| Option | Description |
|---|---|
| statistics | Clears the statistics of all IS-IS interfaces. |

Example: The example below shows how to clear IS-IS interface statistics.

```
supervisor@rtbrick>spine1: op> clear isis interface statistics
```

#### IS-IS Neighbor

**Syntax:**

**clear isis neighbor** <option> ...

| Option | Description |
|---|---|
| `neighbor` | Clears neighbors of the default instance |
| `neighbor instance <instance_name>` | Clears neighbors of the specified instance |
| `neighbor instance <instance_name> interface <interface-name>` | Clears the specified interface of a specified neighbor instance |

Example: The example below shows how to clear neighbors of the specified instance

```
supervisor@rtbrick>spine1: op> clear isis neighbor instance default
```

API To access the Operational State API that corresponds to this CLI, click include::common:partial$operational_api_links.adoc[tag=clear_isis_neighbor].

#### IS-IS Database

**Syntax:**

**clear isis database** <option> ...

| Option | Description |
|---|---|
| `Instance` | Clears neighbors of the default instance |
| `level-1` | Clears area level-1 |
| `level-2` | Clears area level-2 |
| `lsp` | Clears IS-IS database for the specific LSP ID |
| `System` | Clears the LSDB system ID |

Example: The example below shows how to clear the database instance

```
supervisor@rtbrick>spine1: op> clear isis database instance default
```

The LSDB data is cleared for instance default

#### IS-IS Level 2 to Level 1 Route Leak

The command displays the IS-IS Route Leak information.

**show isis database level-1 system** <name>

| Option | Description |
|---|---|
| name | Name of system. |

Example: The output shows the Level-1 ISIS LSP for isr2 is active and advertises both IPv4 and IPv6 reachability along with segment routing (SRGB base 2000, range 1000). The LSP is attached, authenticated with MD5. Most of the prefixes are 'Up' and a few are marked 'Down'.

```
supervisor@rtbrick: op> show isis database level-1 system isr2
Instance: default, Level: 1
  LSP ID: isr2.00-00
    Interface: ifl-0/0/2/0
    LSP Header:
      Sequence: 0x3
      Checksum: 0x1c4
      Remaining lifetime: 65535 seconds
      Flags: Attached: 0, Overload: 0
    Packet:
      Length: 248 bytes
      Last received time: 2026-06-05T14:46:39.411068+0000
      Expiry: expires in 18h 11m 31s 126533us
    System ID: 1980.5110.0121
    Dynamic Hostname TLV (137): isr2
    Protocols Supported TLVs (129):
      Network layer protocol ID: IPv6
      Network layer protocol ID: IPv4
    Area Address TLVs (1):
      Area address: 49.0001
    Authentication TLV (10):
      Type: md5
      Value: 1f82cbc53859f87066a51805990662d7
    IS Reachability TLVs (22):
      IS neighbor: 1980.5110.0111.00              Metric:  1000000
        Adjacency SID:
          Value:      21121, Flags: Value, Local, Persistent
          Value:      21321, Flags: Ipv6 Encapsulation, Value, Local, Persistent
      IS neighbor: 1980.5110.0121.01              Metric:  1000000
    IPv4 Reachability TLVs (135):
      IPv4 prefix: 192.0.2.0/30                   Metric:  1000000   Internal   Up
      IPv4 prefix: 192.0.2.4/30                   Metric:  1000000   Internal   Up
      IPv4 prefix: 198.51.100.21/32               Metric:  1000000   Internal   Up
SID:  200   Flags: Node
      IPv4 prefix: 198.51.100.22/32               Metric:  1000000   Internal   Up
```

```
    IPv6 Reachability TLVs (236):
      IPv6 prefix: 2001:db8:1:1::/64              Metric:  1000000   Internal   Up
      IPv6 prefix: 2001:db8:2:1::/64              Metric:  1000000   Internal   Up
      IPv6 prefix: 2001:db8:0:1::21/128           Metric:  1000000   Internal   Up
SID:  201   Flags: Node
      IPv6 prefix: 2001:db8:0:1::22/128           Metric:  1000000   Internal   Up
    Segment Routing TLVs (242/sub 2):
      SRGB: Base: 11000, Range: 1000, Router ID: 198.51.100.21
  LSP ID: isr2.01-00
    Interface: ifl-0/0/2/0
    LSP Header:
      Sequence: 0x1
      Checksum: 0x2cff
      Remaining lifetime: 65535 seconds
      Flags: Attached: 0, Overload: 0
    Packet:
      Length: 80 bytes
      Last received time: 2026-06-05T14:46:39.411333+0000
      Expiry: expires in 18h 11m 31s 126487us
    System ID: 1980.5110.0121
    Protocols Supported TLVs (129):
      Network layer protocol ID: IPv6
      Network layer protocol ID: IPv4
    Area Address TLVs (1):
      Area address: 49.0001
    Authentication TLV (10):
      Type: md5
      Value: d6d4ece4874d0946847e3e9a81621364
    IS Reachability TLVs (22):
      IS neighbor: 1980.5110.0111.00              Metric:        0
      IS neighbor: 1980.5110.0121.00              Metric:        0
```

Example output for policy configuration that allows only a specific IPv4 route to be leaked from level2 to level1 and blocking the other level 2 route leaking to level 1.

```
supervisor@rtbrick: op> show isis database level-2 system isr2
Instance: default, Level: 2
  LSP ID: isr2.00-00
    Interface: ifl-0/0/2/0
    LSP Header:
      Sequence: 0x6
      Checksum: 0x44fc
      Remaining lifetime: 65535 seconds
      Flags: Attached: 0, Overload: 0
    Packet:
      Length: 342 bytes
      Last received time: 2026-06-05T14:46:40.655342+0000
      Expiry: expires in 18h 11m 31s 324474us
    System ID: 1980.5110.0121
    Dynamic Hostname TLV (137): isr2
```

```
    Protocols Supported TLVs (129):
      Network layer protocol ID: IPv6
      Network layer protocol ID: IPv4
    Area Address TLVs (1):
      Area address: 49.0001
    Authentication TLV (10):
      Type: md5
      Value: cfabe50eef16c90239c4951db3cf2e18
    IS Reachability TLVs (22):
      IS neighbor: 1980.5110.0111.00              Metric:  1000000
        Adjacency SID:
          Value:      21121, Flags: Value, Local, Persistent
          Value:      21321, Flags: Ipv6 Encapsulation, Value, Local, Persistent
      IS neighbor: 1980.5110.0121.01              Metric:  1000000
    IPv4 Reachability TLVs (135):
      IPv4 prefix: 192.0.2.0/30                   Metric:  1000000   Internal   Up
      IPv4 prefix: 192.0.2.4/30                   Metric:  1000000   Internal   Up
      IPv4 prefix: 198.51.100.11/32               Metric:  2000000   Internal   Up
SID:  100   Flags: Re-advertisement, Node
      IPv4 prefix: 198.51.100.12/32               Metric:  2000000   Internal   Up
      IPv4 prefix: 198.51.100.21/32               Metric:  1000000   Internal   Up
SID:  200   Flags: Node
      IPv4 prefix: 198.51.100.22/32               Metric:  1000000   Internal   Up
    IPv6 Reachability TLVs (236):
      IPv6 prefix: 2001:db8:1:1::/64              Metric:  1000000   Internal   Up
      IPv6 prefix: 2001:db8:2:1::/64              Metric:  1000000   Internal   Up
      IPv6 prefix: 2001:db8:0:1::11/128           Metric:  2000000   Internal   Up
SID:  101   Flags: Re-advertisement, Node
      IPv6 prefix: 2001:db8:0:1::12/128           Metric:  2000000   Internal   Up
      IPv6 prefix: 2001:db8:0:1::21/128           Metric:  1000000   Internal   Up
SID:  201   Flags: Node
      IPv6 prefix: 2001:db8:0:1::22/128           Metric:  1000000   Internal   Up
    Segment Routing TLVs (242/sub 2):
      SRGB: Base: 11000, Range: 1000, Router ID: 198.51.100.21
  LSP ID: isr2.01-00
    Interface: ifl-0/0/2/0
    LSP Header:
      Sequence: 0x1
      Checksum: 0x8e6f
      Remaining lifetime: 65535 seconds
      Flags: Attached: 0, Overload: 0
    Packet:
      Length: 80 bytes
      Last received time: 2026-06-05T14:46:40.655203+0000
      Expiry: expires in 18h 11m 31s 324170us
    System ID: 1980.5110.0121
    Protocols Supported TLVs (129):
      Network layer protocol ID: IPv6
      Network layer protocol ID: IPv4
    Area Address TLVs (1):
      Area address: 49.0001
```

```
    Authentication TLV (10):
      Type: md5
      Value: 91ab53688ce2b81fe8df7b09f600acd5
    IS Reachability TLVs (22):
      IS neighbor: 1980.5110.0111.00              Metric:        0
      IS neighbor: 1980.5110.0121.00              Metric:        0
```

# 2.7. OSPFv2/v3

## 2.7.1. OSPF Overview

OSPF (Open Shortest Path First) is an Interior Gateway Protocol that distributes routing information within a single Autonomous System (AS) in an IP network. OSPF is a link-state routing protocol that uses link-state information to form a routing table and exchange the routing information with the neighbors.

RtBrick FullStack (RBFS) supports OSPF version 2 (OSPFv2) and OSPF version 3 (OSPFv3), including authentication, LDP-IGP sync, and redistribution policy. RBFS does not support OSPFv3 Virtual Link.

OSPF routers flood LSAs (link-state advertisements) to all other routers in an autonomous system. Routers generate routing tables using the information received from the LSAs and calculate the best path to other routers in the network. OSPF uses the Dijkstra (Shortest Path First) algorithm to calculate the best path.

LSAs contain local state information such as interfaces and the reachability of neighbors. Other routers, which receive this information as LSAs, build their LSDB (link-state database) using this information. In an OSPF network, all routers build and maintain information about the topology of that network.

**OSPFv3**

OSPF Version 3 (OSPFv3) is a modified version of OSPF and it provides support for the OSPF routing protocol within an IPv6 network. As such, it provides support for IPv6 addresses and prefixes. It retains most of the structure and functions in OSPFv2 (for IPv4) with a few changes.

OSPFv3 differs from OSPFv2 in the following ways:

- OSPFv3 runs on IPv6, which is based on links rather than network segments
- OSPFv3 does not depend on IP addresses
- OSPFv3 packets and the LSA format have the following changes:
  - OSPFv3 router LSAs and network LSAs do not contain IP addresses, which are advertised by Type 8 LSAs and Type 9 LSAs
- In OSPFv3, information about the flooding scope is added in the LSA Type field
  - OSPFv3 stores or floods unidentified packets, whereas OSPFv2 discards
- OSPFv3 supports multi-process on a link with instance ID
- OSPFv3 uses IPv6 link-local addresses for forwarding

#### OSPFv3 Instance ID

One of the advancements of OSPFv3 over OSPFv2 is the use of the instance ID. This instance ID is an 8-bit field within the OSPFv3 header.

The original intent for the instance ID was to support multiple instances of OSPFv3 to run on the same interface. In this way, you can manipulate which routers on a particular segment are allowed to form adjacencies. You could use an instance number of 0 through 255 to distinguish between the different OSPFv3 instances.

However, within RFC 5838, the instance ID was re-purposed to be used to support address families (AFs) with OSPFv3. The default instance of 0 is used if no other instance is defined. However, specific ranges of the instance ID map to specific AFs. According to the RFC, these ranges are:

- Instance ID 0 to 31 — IPv6 unicast AF
- Instance ID 32 to 63 — IPv6 multicast AF
- Instance ID 64 to 95 — IPv4 unicast AF
- Instance ID 96 to 127 — IPv4 multicast AF
- Instance ID 128 to 255 — Unassigned

When using IPv4 unicast or IPv6 unicast, the allowed values for the command are between 0 and 31.

RBFS only supports IPv6 unicast addresses ranging from 0 to 31.

### Understanding OSPF Areas

OSPF allows for a logical partition of the autonomous system by dividing it into areas. This logical partitioning helps to limit the flooding of link-state updates within an area.

An OSPF Autonomous System can be maintained as a single-area network or can be divided as a multi-area network. In a single area AS, the topology provides link-state information of routers in the entire autonomous system.

In a multi-area AS, the topology provides the link-state information of routers belonging to that particular area, not about routers in other areas in the autonomous system. Within an area, all OSPF routers maintain separate databases which are identical.

In a multi-area OSPF network, all areas are connected to the backbone area, known as Area 0.

#### Backbone Area

The backbone area, also known as Area 0, is connected to all other areas in an OSPF network. The backbone area, which acts as a central point of communication, receives LSAs from other areas and disseminates the same to other areas.

#### Area Border Router

Routers that connect one or more areas with the backbone area are called Area Border Router

(ABR). One interface of the ABR is connected to the backbone, while other interfaces are connected to other areas. ABRs, which belong to multiple areas in an OSPF network, maintain separate LSDBs for each area that they are connected to.

The following OSPF architectural diagram shows a simple OSPF network that is divided into areas. Area 1 and Area 2 are connected to the backbone area (Area 0) through the ABRs. Area 1 and Area 2 are not directly connected. They receive link state advertisements from each other from Area 0 which acts as the central point of communication for all other areas.

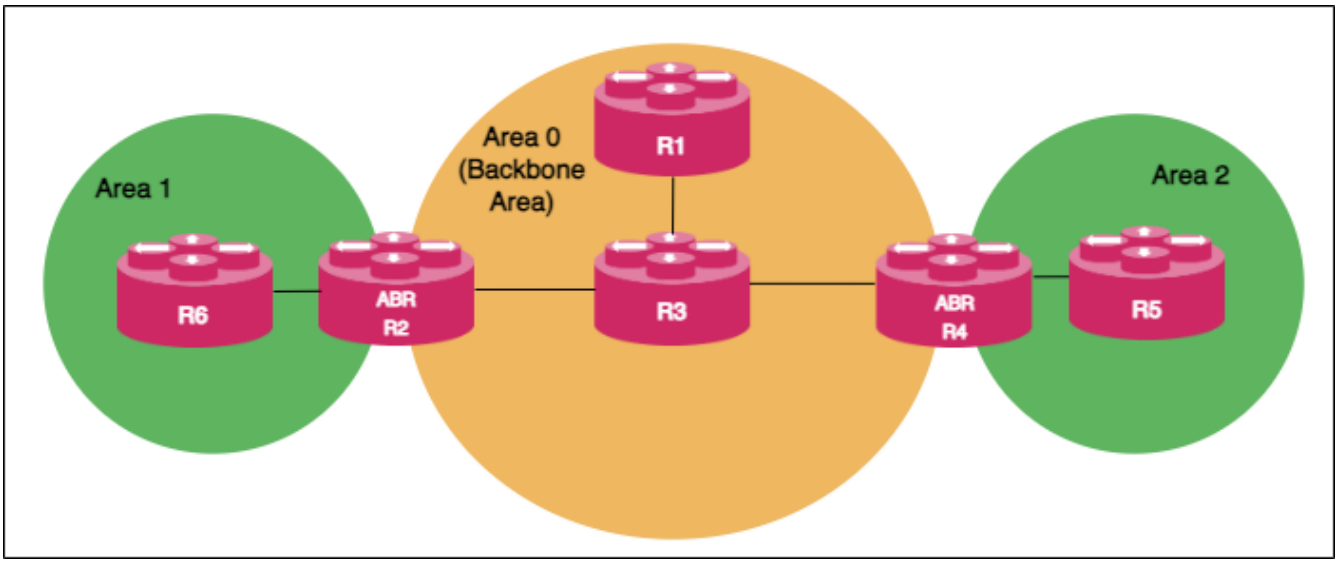



*Figure 16. OSPF Areas*

**Autonomous System Boundary Router**

ASBR (Autonomous System Boundary Router) serves as a gateway router to the OSPF autonomous system. ASBR can operate multiple protocols and work with other autonomous system routers that run other interior gateway protocols such as EIGRP, IS-IS, i-BGP, and so on. ASBR can import and translate different protocol routes into OSPF through the redistribution mechanism.

### OSPF DR and BDR Election

An OSPF network chooses one router as a Designated Router (DR) and another as a Backup Designated Router (BDR) for a broadcast network.

DR acts as a central point of communication by receiving and distributing topology information. BDR takes over the role of DR if the DR fails. Routers in an OSPF network do not directly exchange routing information with each other. Instead, every router in the network updates routing information only with DR and BDR. DR, in turn, distributes the topology information with all other routers. This mechanism reduces network traffic significantly. OSPF chooses one router as DR and another router as BDR based on the following criteria:

- The router with the highest priority value becomes the designated router and the router with the second highest priority value becomes the BDR. You can define the priority values for routers during the interface configuration.
- If multiple routers have the same highest priority value, then the router with the highest router ID is elected as DR and the router with the second highest router ID value becomes the BDR.

You can choose a priority value from the range 0 - 255. Routers with the priority value '0' do not participate in the DR or BDR election.

#### Supported OSPF Standards

RBFS supports the following RFCs, which define standards for OSPFv2 and OSPFv3.

- RFC 2328, OSPF Version 2
- RFC 5340, OSPF for IPv6
- RFC 5709, OSPFv2 HMAC-SHA Cryptographic Authentication
- RFC 7166, Supporting Authentication Trailer for OSPFv3
- RFC 8665, OSPF Extensions for Segment Routing
- RFC 8666, OSPFv3 Extensions for Segment Routing

RFC and draft compliance are partial except as specified.

#### Supported Platforms

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

### 2.7.2. OSPF Configuration

#### Configuration Hierarchy

The diagram illustrates the OSPF configuration hierarchy. All OSPF configuration is performed within an instance, for example, the default instance or a VPN service instance. The OSPF instance configuration hierarchy includes parameters that are generic to the respective OSPF instance. The sub-hierarchies include parameters that are specific to redistribution or authentication.

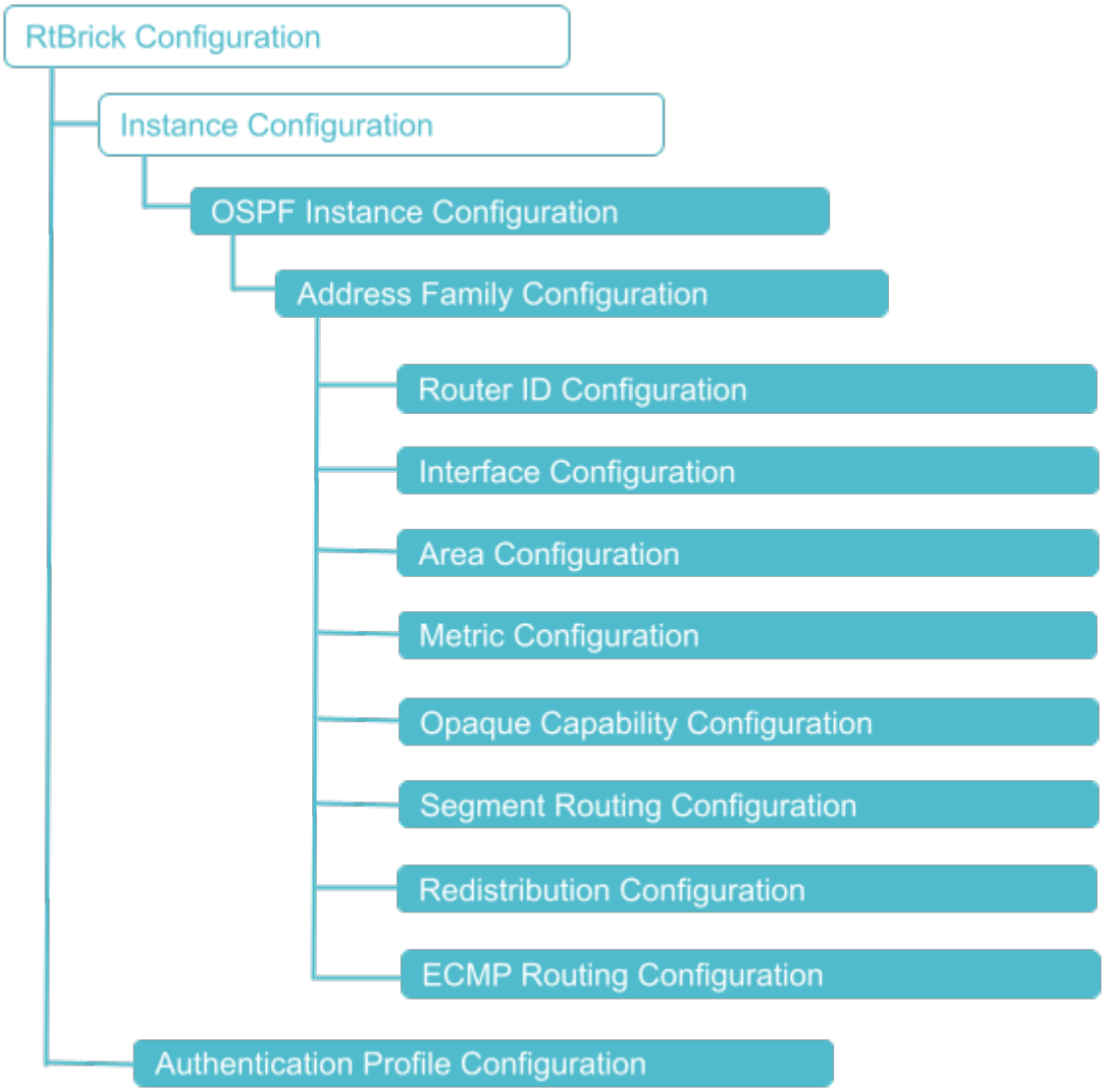



*Figure 17. OSPF Configuration Hierarchy*

### Configuration Syntax and Commands

The following sections describe the OSPF configuration syntax and commands.

#### OSPF Instance Configuration

At this configuration hierarchy, you can configure an OSPF instance.

**Syntax:**

```
set instance <instance-name> protocol ospf
```

| Attribute | Description |
|---|---|
| `<instance-name>` | Name of the OSPF instance. |

API To access the RESTCONF API that corresponds to this CLI, click here.

#### OSPF Address Family Configuration

At this configuration level, you configure the OSPF protocol address family.

> You must configure the OSPF address family on an OSPF instance before configuring other supported OSPF features.

### Syntax

**set instance** <instance-name> **protocol ospf address-family** <afi>

| Attribute | Description |
|---|---|
| `<instance-name>` | Name of the instance |
| `<afi>` | Specifies the Address family identifier (AFI), either IPv4 or IPv6. |

The following example shows the OSPF address family (IPv4) configuration.

Example: OSPF Instance Address Family IPv4 Configuration

This configuration sets up the IPv4 address family for the OSPF protocol. It defines router ID, OSPF area (0.0.0.0), and the interfaces (ifl-0/0/0/1 and lo-0/0/0/1) associated with that area and specifies a point-to-point network type for `ifl-0/0/0/1`.

```
set instance default protocol ospf address-family ipv4
set instance default protocol ospf address-family ipv4 router-id 198.51.100.10
set instance default protocol ospf address-family ipv4 area 0.0.0.0
set instance default protocol ospf address-family ipv4 area 0.0.0.0 interface ifl-
0/0/0/1
set instance default protocol ospf address-family ipv4 area 0.0.0.0 interface ifl-
0/0/0/1 network-type p2p
set instance default protocol ospf address-family ipv4 area 0.0.0.0 interface lo-
0/0/0/1
```

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ospf address-family ipv4
{
  "rtbrick-config:ipv4": {
    "router-id": "198.51.100.10",
    "area": [
      {
        "area-id": "0.0.0.0",
        "interface": [
          {
            "name": "ifl-0/0/0/1",
            "network-type": "p2p"
          },
          {
            "name": "lo-0/0/0/1"
```

```
          }
        ]
      }
    ]
  }
}
<...>
```

Example: OSPF Instance Address Family IPv6 Configuration

The following example shows the OSPF IPv6 address family configuration. This configuration sets up an OSPF instance for the IPv6 address family. It defines a router ID, a default metric, and enables segment routing with a specified SRGB. It also configures route redistribution from BGP and direct routes using the policy named 'ospf_policy_1'. It also configures an OSPF area '0.0.0.0' and lists a few interfaces (ifl-0/0/0/1 , ifl-0/0/0/100, ifl-0/0/1/1, and ifl-0/0/1/100) within this area, each with a specific metric, network type (where applicable), and an authentication profile named 'AUTH_PROFILE_SHA_512_1'.

```
set instance default protocol ospf address-family ipv6 instance-id 0 router-id
192.168.0.10
set instance default protocol ospf address-family ipv6 instance-id 0 metric 1000
set instance default protocol ospf address-family ipv6 instance-id 0 segment-routing
status enable
set instance default protocol ospf address-family ipv6 instance-id 0 segment-routing
srgb base 1000
set instance default protocol ospf address-family ipv6 instance-id 0 segment-routing
srgb range 1000
set instance default protocol ospf address-family ipv6 instance-id 0 redistribute
source bgp metric 2000
set instance default protocol ospf address-family ipv6 instance-id 0 redistribute
source direct policy ospf_policy_1
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
metric 1000
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
interface ifl-0/0/0/1 authentication-profile AUTH_PROFILE_SHA_512_1
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
interface ifl-0/0/0/100 metric 20000
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
interface ifl-0/0/0/100 network-type p2p
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
interface ifl-0/0/0/100 authentication-profile AUTH_PROFILE_SHA_512_1
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
interface ifl-0/0/1/1 metric 40000
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
interface ifl-0/0/1/1 network-type p2p
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
interface ifl-0/0/1/1 authentication-profile AUTH_PROFILE_SHA_512_1
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
interface ifl-0/0/1/100 metric 30000
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
```

```
interface ifl-0/0/1/100 authentication-profile AUTH_PROFILE_SHA_512_1
```

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ospf address-family ipv6
{
  "rtbrick-config:ipv6": {
    "instance-id": [
      {
        "instance-id": 0,
        "router-id": "192.168.0.10",
        "metric": 1000,
        "segment-routing": {
          "status": "enable",
          "srgb": {
            "base": 1000,
            "range": 1000
          }
        },
        "redistribute": [
          {
            "source": "bgp",
            "metric": 2000
          },
          {
            "source": "direct",
            "policy": "ospf_policy_1"
          }
        ],
        "area": [
          {
            "area-id": "0.0.0.0",
            "metric": 1000,
            "interface": [
              {
                "name": "ifl-0/0/0/1",
                "authentication-profile": "AUTH_PROFILE_SHA_512_1"
              },
              {
                "name": "ifl-0/0/0/100",
                "metric": 20000,
                "network-type": "p2p",
                "authentication-profile": "AUTH_PROFILE_SHA_512_1"
              },
              {
                "name": "ifl-0/0/1/1",
                "metric": 40000,
                "network-type": "p2p",
                "authentication-profile": "AUTH_PROFILE_SHA_512_1"
              },
              {
                "name": "ifl-0/0/1/100",
                "metric": 30000,
                "authentication-profile": "AUTH_PROFILE_SHA_512_1"
<...>
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### OSPF Router ID Configuration

The router ID is an IP address that OSPF uses to identify a device on the network. The router ID should be configured under the address family hierarchy.

### Syntax

**set instance** <instance-name> **protocol ospf address-family** <afi> <attribute> <value>

| Attribute | Description |
|---|---|
| `<afi>` | Specifies the Address family identifier (AFI), either IPv4 or IPv6. |
| `<afi> instance-id <instance-id>` | Specifies the instance ID. When using an IPv6 address family with OSPFv3, an instance ID ranging from 0 to 31 must be specified. |
| `router-id <ipv4-address>` | The router ID of the routing instance. It is recommended to specify the router ID. |

Example: OSPF IPv4 Router Identifier Configuration

This command configures router ID for a router that belong to the OSPF IPv4 address family.

```
set instance default protocol ospf address-family ipv4 router-id 192.168.0.10
```

```
supervisor@rtbrick: cfg> show config instance default protocol ospf address-family
ipv4 router-id
{
  "rtbrick-config:router-id": "192.168.0.10"
}
```

Example: OSPF IPv6 Router Identifier Configuration

This command configures router ID for OSPF the IPv6 address family. It sets the router's unique identifier (192.168.0.10) for the OSPF instance 0.

```
set instance default protocol ospf address-family ipv6 instance-id 0 router-id
192.168.0.10
```

```
supervisor@rtbrick: cfg> show config instance default protocol ospf address-family
ipv6 instance-id 0 router-id
{
  "rtbrick-config:router-id": "192.168.0.10"
```

```
}
```

#### OSPF Hostname Configuration

Identifying routers through Open Shortest Path First (OSPF) hostnames can simplify network management as they are easier to remember than router IDs.

### Syntax

**set instance** <instance-name> **protocol ospf address-family** <afi> <attribute> <value>

| Attribute | Description |
|---|---|
| `<afi>` | Specifies the Address family identifier (AFI), either IPv4 or IPv6. |
| `<afi> instance-id <instance-id>` | Specifies the instance ID. When using an IPv6 address family with OSPFv3, an instance ID ranging from 0 to 31 must be specified. |
| `hostname <hostname>` | The hostname of the routing instance. |

Example: OSPF IPv4 host name Configuration

This command sets the OSPF IPv4 default instance host name to 'multiservice-edge'.

```
set instance default protocol ospf address-family ipv4 hostname multiservice-edge
```

```
supervisor@rtbrick: cfg> show config instance default protocol ospf address-family
ipv4 hostname
{
  "rtbrick-config:hostname": "multiservice-edge"
}
```

Example: OSPF IPv6 host name configuration

This command sets the OSPF IPv6 default instance host name to 'multiservice-edge'.

```
set instance default protocol ospf address-family ipv6 hostname multiservice-edge
```

```
supervisor@rtbrick: cfg> show config instance default protocol ospf address-family
ipv6 instance-id 0 hostname
{
  "rtbrick-config:hostname": "multiservice-edge"
}
```

#### OSPF Area Configuration

A particular area is defined by its area ID.

**set instance** <instance-name> **protocol ospf address-family** <afi> <attribute> <value>

| Attribute | Description |
|---|---|
| `<afi>` | Specifies the Address family identifier (AFI), either IPv4 or IPv6. |
| `<afi> instance-id <instance-id>` | Specifies the instance ID. When using an IPv6 address family with OSPFv3, an instance ID ranging from 0 to 31 must be specified. |
| `area <area-id>` | Specifies the OSPF area ID. |
| `area <area-id> metric` | Area scope metric. Range: 0 - 65535. Default: 10000. |
| `area <area-id> area-type stub` | A stub area is an area through which or into which AS external advertisements are not flooded instead type-3 LSA advertise with a default route. |
| `area <area-id> area-type totally-stub` | Totally stub area is an area in which type-3 LSAs are not allowed instead type-3 LSAs advertise with a default route. |
| `area <area-id> authentication-profile <authentication-profile>` | Specifies the authentication profile name used to create an attachment point at the area level. |
| `area <area-id> no-authentication-check <enable>` | When enabled, OSPF packets received here will not undergo authentication validation, even if the user has enabled authentication. However, OSPF will continue to send authenticated packets from this interface. |

Example: OSPF IPv4 Area Configuration

The following configuration defines OSPFv2 settings for IPv4 under the default instance and associates multiple interfaces with area 0.0.0.0. Each interface is configured with parameters such as metric, network type, and authentication profile. This configuration includes multiple physical and logical interfaces connected to OSPF area 0.0.0.0, with varying interface metrics to manage path preference effectively. For the interface 'ifl-0/0/0/100', the metric is set to 20,000; for 'ifl-0/0/1/1', the metric is set to 40,000; and for 'ifl-0/0/4/1', the metric is set to 60,000. This defines the interface cost, which influences OSPF route selection, the lower the metric, the more preferred the path.

The network type is configured as point-to-point (p2p), indicating that OSPF considers this link as a direct connection between two routers. This point-to-point network type is applied to both the interfaces 'ifl-0/0/1/100' and 'ifl-0/0/1/1'. Also, the loopback interfaces 'lo-0/0/0/1' and 'lo-0/0/0/2' are included in OSPF area '0.0.0.0'.

```
set instance default protocol ospf address-family ipv4 area 0.0.0.0 interface ifl-
0/0/0/1 authentication-profile AUTH_PROFILE_SHA_512_1
set instance default protocol ospf address-family ipv4 area 0.0.0.0 interface ifl-
0/0/0/100 metric 20000
```

```
set instance default protocol ospf address-family ipv4 area 0.0.0.0 interface ifl-
0/0/0/100 network-type p2p
set instance default protocol ospf address-family ipv4 area 0.0.0.0 interface ifl-
0/0/0/100 authentication-profile AUTH_PROFILE_SHA_512_1
set instance default protocol ospf address-family ipv4 area 0.0.0.0 interface ifl-
0/0/1/1 metric 40000
set instance default protocol ospf address-family ipv4 area 0.0.0.0 interface ifl-
0/0/1/1 network-type p2p
set instance default protocol ospf address-family ipv4 area 0.0.0.0 interface ifl-
0/0/1/1 authentication-profile AUTH_PROFILE_SHA_512_1
set instance default protocol ospf address-family ipv4 area 0.0.0.0 interface ifl-
0/0/1/100 metric 30000
set instance default protocol ospf address-family ipv4 area 0.0.0.0 interface ifl-
0/0/1/100 authentication-profile AUTH_PROFILE_SHA_512_1
set instance default protocol ospf address-family ipv4 area 0.0.0.0 interface ifl-
0/0/4/1 metric 60000
set instance default protocol ospf address-family ipv4 area 0.0.0.0 interface lo-
0/0/0/1
set instance default protocol ospf address-family ipv4 area 0.0.0.0 interface lo-
0/0/0/2
```

```
supervisor@rtbrick: cfg> show config instance default protocol ospf address-family ipv4 area 0.0.0.0
{
  "rtbrick-config:area": [
    {
      "area-id": "0.0.0.0",
      "interface": [
        {
          "name": "ifl-0/0/0/1",
          "authentication-profile": "AUTH_PROFILE_SHA_512_1"
        },
        {
          "name": "ifl-0/0/0/100",
          "metric": 20000,
          "network-type": "p2p",
          "authentication-profile": "AUTH_PROFILE_SHA_512_1"
        },
        {
          "name": "ifl-0/0/1/1",
          "metric": 40000,
          "network-type": "p2p",
          "authentication-profile": "AUTH_PROFILE_SHA_512_1"
        },
        {
          "name": "ifl-0/0/1/100",
          "metric": 30000,
          "authentication-profile": "AUTH_PROFILE_SHA_512_1"
        },
        {
          "name": "ifl-0/0/4/1",
          "metric": 60000
        },
        {
```

```
          "name": "lo-0/0/0/1"
        },
        {
          "name": "lo-0/0/0/2"
        }
      ]
    }
  ]
}
```

Example: OSPF IPv6 Area Configuration

This configuration defines OSPFv3 (IPv6) settings under the default instance for area 0.0.0.0. Multiple interfaces are associated with this area, each configured with parameters such as metric, network type, and authentication profile. The area includes interfaces under the instance 'default' and instance-id 0. Each interface is configured with parameters such as metric, network type, and authentication profile. It uses the authentication profile 'AUTH_PROFILE_SHA_512_1' for secure OSPF authentication for interfaces 'ifl-0/0/0/1', 'ifl-0/0/0/100', 'ifl-0/0/1/1', and 'ifl-0/0/1/100'.

It configures OSPF settings for this instance, including specific parameters such as metrics, network types, and authentication requirements for the interfaces. Interfaces 'ifl-0/0/0/100' and 'ifl-0/0/1/1' are configured as point-to-point links. These interfaces also have custom OSPF metrics set to influence route selection: 20,000 and 40,000 respectively. The interface 'ifl-0/0/1/100' uses the same authentication profile and has a metric of 30,000. The interface 'ifl-0/0/4/1' is configured with a metric of 60,000 but does not employ authentication. The loopback interfaces 'lo-0/0/0/1' and 'lo-0/0/0/2' are included in the OSPF configuration.

```
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
interface ifl-0/0/0/1 authentication-profile AUTH_PROFILE_SHA_512_1
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
interface ifl-0/0/0/100 metric 20000
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
interface ifl-0/0/0/100 network-type p2p
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
interface ifl-0/0/0/100 authentication-profile AUTH_PROFILE_SHA_512_1
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
interface ifl-0/0/1/1 metric 40000
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
interface ifl-0/0/1/1 network-type p2p
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
interface ifl-0/0/1/1 authentication-profile AUTH_PROFILE_SHA_512_1
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
interface ifl-0/0/1/100 metric 30000
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
interface ifl-0/0/1/100 authentication-profile AUTH_PROFILE_SHA_512_1
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
interface ifl-0/0/4/1 metric 60000
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
interface lo-0/0/0/1
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
```

```
interface lo-0/0/0/2
```

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ospf address-family ipv6 instance-id 0 area
0.0.0.0
{
  "rtbrick-config:area": [
    {
      "area-id": "0.0.0.0",
      "interface": [
        {
          "name": "ifl-0/0/0/1",
          "authentication-profile": "AUTH_PROFILE_SHA_512_1"
        },
        {
          "name": "ifl-0/0/0/100",
          "metric": 20000,
          "network-type": "p2p",
          "authentication-profile": "AUTH_PROFILE_SHA_512_1"
        },
        {
          "name": "ifl-0/0/1/1",
          "metric": 40000,
          "network-type": "p2p",
          "authentication-profile": "AUTH_PROFILE_SHA_512_1"
        },
        {
          "name": "ifl-0/0/1/100",
          "metric": 30000,
          "authentication-profile": "AUTH_PROFILE_SHA_512_1"
        },
        {
          "name": "ifl-0/0/4/1",
          "metric": 60000
        },
        {
          "name": "lo-0/0/0/1"
        },
        {
          "name": "lo-0/0/0/2"
        }
      ]
    }
  ]
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### OSPF Interface Configuration

Enable OSPF protocol on the router interfaces.

**Syntax:**

**set instance** <instance-name> **protocol ospf address-family** <afi> <attribute> <value>

| Attribute | Description |
|---|---|
| `<afi>` | Specifies the Address family identifier (AFI), either IPv4 or IPv6. |

| Attribute | Description |
| --- | --- |
| `<afi> instance-id <instance-id>` | Specifies the instance ID. When using an IPv6 address family with OSPFv3, an instance ID ranging from 0 to 31 must be specified. |
| `area <area-id> interface <interface-name> ldp-synchronization enable` | Enables LDP OSPF Synchronization. By default, LDP OSPF synchronization is disabled. |
| `metric <metric>` | Specify the metric value of an OSPF interface. |
| `multipath` | Specifies the maximum ECMP paths that can be calculated.<br><br>The default ECMP Maximum Next-hop Count is limited to 8; however, this limit varies depending on the platform that RBFS supports. For details on the limits set for the ECMP Maximum Next-hop Count on the platforms, refer to the Resource Limits/Feature Support section of the *Platform Guide*. |
| `network-type <broadcast \| p2p>` | broadcast - Sets the network type to broadcast; p2p - Sets the network type to point-to-point. By default, the network-type is `broadcast`. |
| `router-priority <router-priority>` | Sets the router priority for an interface. Allowed range: 0 - 255, Default: 1. Routers with priority value '0' do not participate in the DR or BDR election. |
| `segment-routing index` | Sets the prefix segment identifier (SID) index for the specified interface. |
| `timer <hello \| dead \| wait>` | Interface timer for configuring `hello`, `dead`, and `wait` timers.<br><br>• **hello**: Sets interval time for sending hello packets to a neighbor and this time is identical on OSPF neighbor routers. Range: 1-65535 seconds. Default: 10 seconds.<br>• **dead**: Sets interval time within which if the interface does not receive any hello packet from its neighbor, the interface comes to know that the neighbor is down. Range: 4-262140 seconds. Default: 40 seconds.<br>• **wait**: Sets wait time to delay to trigger the DR/BDR election. It cannot be more than the time set for the dead interval. Range: 1-255 seconds. Default: 40 seconds. |
| `mtu-ignore enable` | If there is an MTU mismatch on both sides of the link where OSPF runs, the OSPF adjacency will not come up as the MTU value carried in the Database Description (DBD) packets. To avoid MTU validation in the Database Description (DBD) packets, configure `mtu-ignore` command. By default, it is disabled. |

| Attribute | Description |
| --- | --- |
| `authentication <authentication-profile>` | Specifies the authentication profile name used to create an attachment point at the interface level. |
| `no-authentication-check <enable>` | When enabled, OSPF packets received here will not undergo authentication validation at the interface level, even if the user has enabled authentication. |

If an authentication profile is attached to an interface and an area, the authentication profile attached to the interface takes priority.

Example: OSPF IPv4 Interface Configuration

The following configuration sets up interfaces in the OSPF protocol under the default OSPF instance, within the IPv4 address family and area 0.0.0.0. The interfaces 'ifl-0/0/0/1', 'ifl-0/0/0/100', 'ifl-0/0/1/1', and 'ifl-0/0/1/100' are defined and they use the same authentication profile 'AUTH_PROFILE_SHA_512_1'.

The interfaces 'ifl-0/0/0/100' and 'ifl-0/0/1/1' are configured as point-to-point links. These interfaces also have custom OSPF metrics set to influence route selection: 20,000 and 40,000 respectively. The interface 'ifl-0/0/1/100' uses the same authentication profile and has a metric of 30,000, although it does not have a specific network type defined. Additionally, the interface 'ifl-0/0/4/1' is configured with a metric of 60,000 but does not employ authentication. The loopback interfaces 'lo-0/0/0/1' and 'lo-0/0/0/2' are included in the OSPF configuration with default settings.

```
set instance default protocol ospf address-family ipv4 area 0.0.0.0 interface ifl-
0/0/0/1 authentication-profile AUTH_PROFILE_SHA_512_1
set instance default protocol ospf address-family ipv4 area 0.0.0.0 interface ifl-
0/0/0/100 metric 20000
set instance default protocol ospf address-family ipv4 area 0.0.0.0 interface ifl-
0/0/0/100 network-type p2p
set instance default protocol ospf address-family ipv4 area 0.0.0.0 interface ifl-
0/0/0/100 authentication-profile AUTH_PROFILE_SHA_512_1
set instance default protocol ospf address-family ipv4 area 0.0.0.0 interface ifl-
0/0/1/1 metric 40000
set instance default protocol ospf address-family ipv4 area 0.0.0.0 interface ifl-
0/0/1/1 network-type p2p
set instance default protocol ospf address-family ipv4 area 0.0.0.0 interface ifl-
0/0/1/1 authentication-profile AUTH_PROFILE_SHA_512_1
set instance default protocol ospf address-family ipv4 area 0.0.0.0 interface ifl-
0/0/1/100 metric 30000
set instance default protocol ospf address-family ipv4 area 0.0.0.0 interface ifl-
0/0/1/100 authentication-profile AUTH_PROFILE_SHA_512_1
set instance default protocol ospf address-family ipv4 area 0.0.0.0 interface ifl-
0/0/4/1 metric 60000
set instance default protocol ospf address-family ipv4 area 0.0.0.0 interface lo-
0/0/0/1
set instance default protocol ospf address-family ipv4 area 0.0.0.0 interface lo-
```

```
0/0/0/2
```

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ospf address-family ipv4 area 0.0.0.0 interface
{
  "rtbrick-config:interface": [
    {
      "name": "ifl-0/0/0/1",
      "authentication-profile": "AUTH_PROFILE_SHA_512_1"
    },
    {
      "name": "ifl-0/0/0/100",
      "metric": 20000,
      "network-type": "p2p",
      "authentication-profile": "AUTH_PROFILE_SHA_512_1"
    },
    {
      "name": "ifl-0/0/1/1",
      "metric": 40000,
      "network-type": "p2p",
      "authentication-profile": "AUTH_PROFILE_SHA_512_1"
    },
    {
      "name": "ifl-0/0/1/100",
      "metric": 30000,
      "authentication-profile": "AUTH_PROFILE_SHA_512_1"
    },
    {
      "name": "ifl-0/0/4/1",
      "metric": 60000
    },
    {
      "name": "lo-0/0/0/1"
    },
    {
      "name": "lo-0/0/0/2"
    }
  ]
}
```

Example: OSPF IPv6 Interface Configuration

This configuration defines OSPFv3 (IPv6) interfaces under the instance 'default', instance ID '0', and the area 0.0.0.0. Each interface is configured with parameters such as metric, network type, and authentication profile. It uses the authentication profile AUTH_PROFILE_SHA_512_1 for interfaces 'ifl-0/0/0/1', 'ifl-0/0/0/100', 'ifl-0/0/1/1', and 'ifl-0/0/1/100'.

Interfaces 'ifl-0/0/0/100' and 'ifl-0/0/1/1' are configured as point-to-point links. These interfaces also have custom OSPF metrics set to influence route selection: 20,000 and 40,000 respectively. The interface 'ifl-0/0/1/100' uses the same authentication profile and has a metric of 30,000. The interface 'ifl-0/0/4/1' is configured with a metric of 60,000 but does not employ authentication. The loopback interfaces 'lo-0/0/0/1' and 'lo-0/0/0/2' are included in the OSPF configuration with default settings.

```
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
interface ifl-0/0/0/1 authentication-profile AUTH_PROFILE_SHA_512_1
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
```

```
interface ifl-0/0/0/100 metric 20000
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
interface ifl-0/0/0/100 network-type p2p
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
interface ifl-0/0/0/100 authentication-profile AUTH_PROFILE_SHA_512_1
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
interface ifl-0/0/1/1 metric 40000
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
interface ifl-0/0/1/1 network-type p2p
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
interface ifl-0/0/1/1 authentication-profile AUTH_PROFILE_SHA_512_1
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
interface ifl-0/0/1/100 metric 30000
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
interface ifl-0/0/1/100 authentication-profile AUTH_PROFILE_SHA_512_1
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
interface ifl-0/0/4/1 metric 60000
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
interface lo-0/0/0/1
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
interface lo-0/0/0/2
```

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ospf address-family ipv6 instance-id 0 area
0.0.0.0 interface
{
  "rtbrick-config:interface": [
    {
      "name": "ifl-0/0/0/1",
      "authentication-profile": "AUTH_PROFILE_SHA_512_1"
    },
    {
      "name": "ifl-0/0/0/100",
      "metric": 20000,
      "network-type": "p2p",
      "authentication-profile": "AUTH_PROFILE_SHA_512_1"
    },
    {
      "name": "ifl-0/0/1/1",
      "metric": 40000,
      "network-type": "p2p",
      "authentication-profile": "AUTH_PROFILE_SHA_512_1"
    },
    {
      "name": "ifl-0/0/1/100",
      "metric": 30000,
      "authentication-profile": "AUTH_PROFILE_SHA_512_1"
    },
    {
      "name": "ifl-0/0/4/1",
      "metric": 60000
    },
    {
      "name": "lo-0/0/0/1"
    },
    {
      "name": "lo-0/0/0/2"
    }
  ]
```

```
}
```

To access the RESTCONF API that corresponds to this CLI, click here.

#### OSPF Metric Configuration

Metric is the cost that OSPF uses to calculate and identify the best paths to other routers.

### Syntax

**set instance** <instance-name> **protocol ospf address-family** <afi> <attribute> <value>

| Attribute | Description |
|---|---|
| `<afi>` | Specifies the Address family identifier (AFI), either IPv4 or IPv6. |
| `<afi> instance-id <instance-id>` | Specifies the instance ID. When using an IPv6 address family with OSPFv3, an instance ID ranging from 0 to 31 must be specified. |
| `metric <metric>` | OSPF address-family metric. Allowed range: 0 - 65535. Default: 10000.<br>If you configure the metric at the address family, it will be applicable to the configured areas of the address-family. If you configure a metric for an area, this configured metric value will take precedence over the address-family metric configurations of this area.<br>If you specify a metric value for an area on an interface will override any area and address-family metric configurations for this area. |

Example 1: OSPF IPv4 Metric Configuration

This configuration sets the OSPF cost (metric) for all interfaces within area 0.0.0.0 (for IPv4 address family) to 1000.

```
set instance default protocol ospf address-family ipv4 area 0.0.0.0 metric 1000
```

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ospf address-family ipv4 area 0.0.0.0 metric
{
  "rtbrick-config:metric": 1000
}
```

Example: OSPF IPv6 Metric Configuration

This configuration sets the OSPFv3 cost (metric) for all interfaces within area 0.0.0.0 (for IPv6 address family) to 1000.

```
set instance default protocol ospf address-family ipv6 instance-id 0  area 0.0.0.0
metric 1000
```

```
supervisor@rtbrick: cfg> show config instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0 metric
{
  "rtbrick-config:metric": 1000
}
```

#### OSPF Opaque Capability Configuration

Enables opaque link-state advertisements. Routers in the OSPF network can receive and advertise Type-9, Type-10 and Type-11 opaque LSAs.

### Syntax

**set instance** <instance-name> **protocol ospf address-family ipv4** <attribute> <value>

| Attribute | Description |
|---|---|
| `opaque-capability <enable \| disable>` | Enable or disable opaque LSA advertisement and reception. Set as 'enable' to enable the router to receive and advertise opaque LSAs. |

Example: OSPF Opaque Capability Configuration

This configuration enables opaque capability for the OSPF IPv4 address family. This configuration allows the exchange of opaque LSAs.

```
set instance default protocol ospf address-family ipv4 opaque-capability enable
```

```
supervisor@rtbrick>: cfg> show config instance default protocol ospf address-family ipv4 opaque-capability
{
  "rtbrick-config:opaque-capability": "enable"
}
```

#### Segment Routing Configuration

It enables segment routing for OSPF. For configuring segment routing, you must enable the opaque capability by defining it as 'true'. For information, see the section: "Opaque Capability Configuration".

**Syntax**

**set instance** <instance-name> **protocol ospf address-family** <afi> <attribute> <value>

| Attribute | Description |
|---|---|
| `<afi>` | Specifies the Address family identifier (AFI), either IPv4 or IPv6. |
| `<afi> instance-id <instance-id>` | Specifies the instance ID. When using an IPv6 address family with OSPFv3, an instance ID ranging from 0 to 31 must be specified. |
| `segment-routing srgb base <value>` | Specifies the segment routing global block (SRGB) in source packet routing. SRGB is used for prefix SIDs. |
| `segment-routing srgb range <value>` | OSPF system range of labels from the base label.<br><br>NOTE: For details on the supported label ranges, see the Static, System, and Dynamic Label Ranges section in the Platform Guide. |
| `segment-routing status <disable \| enable>` | Enable or disable the segment routing feature. By default, the status is disabled. |

Example: OSPF IPv4 Segment routing Configuration

This configuration enables Segment Routing for the OSPF IPv4 address family. It defines SRGB base as '1000' and the range as '1000'.

```
set instance default protocol ospf address-family ipv4 segment-routing status enable
set instance default protocol ospf address-family ipv4 segment-routing srgb base 1000
set instance default protocol ospf address-family ipv4 segment-routing srgb range 1000
```

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ospf address-family ipv4 segment-routing
{
  "rtbrick-config:segment-routing": {
    "status": "enable",
    "srgb": {
      "base": 1000,
      "range": 1000
    }
  }
}
```

Example: OSPF IPv6 Segment routing Configuration

This configuration enables Segment Routing for the OSPF IPv6 address family. It defines SRGB base as '1000' and the range as '1000'.

```
set instance default protocol ospf address-family ipv6 instance-id 0 segment-routing
status enable
```

```
set instance default protocol ospf address-family ipv6 instance-id 0 segment-routing
srgb base 1000
set instance default protocol ospf address-family ipv6 instance-id 0 segment-routing
srgb range 1000
```

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ospf address-family ipv6 instance-id 0 segment-
routing
{
  "rtbrick-config:segment-routing": {
    "status": "enable",
    "srgb": {
      "base": 1000,
      "range": 1000
    }
  }
}
```

API To access the RESTCONF API that corresponds to this CLI (OSPF IPv4 Segment routing), click here.

API To access the RESTCONF API that corresponds to this CLI (OSPF IPv6 Segment routing), click here.

#### OSPF Redistribution Configuration

It enables route redistribution for the routes originating from other sources or protocols such as `BGP`, `Direct`, `IPoE`, `IS-IS`, `PPP`, and `Static`.

### Syntax

**set instance** <instance-name> **protocol ospf** <afi> <attribute> <value>

| Attribute | Description |
|---|---|
| `<afi>` | Specifies the Address family identifier (AFI), either IPv4 or IPv6. |
| `<afi> instance-id <instance-id>` | Specifies the instance ID. When using an IPv6 address family with OSPFv3, an instance ID ranging from 0 to 31 must be specified. |
| `redistribute <protocol>` | Specifies the source protocol from which the routes are to be redistributed. The available options include `BGP`, `Direct`, `IPoE`, `IS-IS`, `PPP`, and `Static`. |
| `metric <metric>` | Specifies the metric value for the redistributed routes. |
| `metric-type <type 1 \| type 2>` | Specifies the external metric type for the redistributed routes. |
| `policy` | Specifies the name of the policy map. The redistribute attach point allows routes from other sources to be advertised by OSPFv2. |

Example: OSPF IPV4 BGP Redistribution Configuration

The following OSPF IPv4 redistribution configuration ensures that BGP-learned routes are injected into OSPF with a defined metric (2000). It ensures routing across protocol boundaries.

```
set instance default protocol ospf address-family ipv4 redistribute bgp
set instance default protocol ospf address-family ipv4 redistribute bgp metric 2000
```

```
supervisor@rtbrick: cfg> show config instance default protocol ospf address-family ipv4 redistribute bgp
{
  "rtbrick-config:redistribute": [
    {
      "source": "bgp",
      "metric": 2000
    }
  ]
}
```

Example: OSPF IPv6 Redistribution Policy

The following IPv6 redistribution policy configuration ensures that only the directly connected routes are allowed to import into OSPFv3 according to the rules defined in ospf_policy_1.

```
set instance default protocol ospf address-family ipv6 instance-id 0 redistribute
source direct policy ospf_policy_1
```

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ospf address-family ipv6 instance-id 0
redistribute direct
{
  "rtbrick-config:redistribute": [
    {
      "source": "direct",
      "policy": "ospf_policy_1"
    }
  ]
}
```

API To access the RESTCONF API that corresponds to this CLI (OSPF IPv4 Redistribution), click here.

API To access the RESTCONF API that corresponds to this CLI (OSPF IPv6 Redistribution ), click here.

#### ECMP Routing Configuration

ECMP (equal-cost multiple paths) routing is a mechanism in which routers forward packets to a destination using the multiple available best paths. This mechanism can increase network bandwidth substantially by load-balancing traffic through multiple best paths.

**Syntax**

**set instance** <instance-name> **protocol ospf address-family** <afi> <attribute> <value>

| Attribute | Description |
|---|---|
| `<afi>` | Specifies the Address family identifier (AFI), either IPv4 or IPv6. |
| `<afi> instance-id <instance-id>` | Specifies the instance ID. When using an IPv6 address family with OSPFv3, an instance ID ranging from 0 to 31 must be specified. |
| `<multipath>` | Maximum ECMP paths to be calculated. |

Example: OSPF IPv4 ECMP Routing Configuration

This configuration sets OSPF as the default protocol for IPv4 address-family and enables up to two equal-cost multipath routes.

```
set instance default protocol ospf address-family ipv4 multipath 2
```

```
supervisor@rtbrick: cfg> show config instance default protocol ospf address-family ipv4 multipath
{
  "rtbrick-config:multipath": 2
}
```

Example: OSPF IPv6 ECMP Routing Configuration

This configuration sets OSPF as the default protocol for the IPv6 address-family with instance ID 0 and allows up to two equal-cost multipath routes.

```
set instance default protocol ospf address-family ipv6 instance-id 0 multipath 2
```

```
supervisor@rtbrick: cfg> show config instance default protocol ospf address-family ipv6 instance-id 0 multipath
{
  "rtbrick-config:multipath": 2
}
```

#### OSPF Authentication Configuration

OSPF supports the secure exchange of routing updates through authentication. You can enable authentication by attaching an authentication profile at the area or interface level. OSPF allows multiple keys to be attached to prevent session interruption.

The table below shows the authentication types supported by OSPFv2 and v3.

*OSPF Authentication Types Supported*

| Authentication Type | OSPFv2 | OSPFv3 |
|---|---|---|
| `Clear Text` | Yes | No |
| `HMAC-SHA-1` | Yes | Yes |
| `HMAC-SHA-256` | Yes | Yes |
| `HMAC-SHA-384` | Yes | Yes |
| `HMAC-SHA-512` | Yes | Yes |
| `MD5` | Yes | No |

To authenticate OSPF, there must be a global authentication profile present.

**Configuring an Authentication Profile**

**set authentication-profile** <attribute> <value>

| Attribute | Description |
|---|---|
| `<name>` | Specifies the authentication profile name. |
| `<name> key <key-id>` | Specifies the message digest key identifier to be used by the neighboring routers for the OSPF password authentication. Allowed range: 1 - 255. |
| `<name> key <key-id> type <auth-type>` | Specifies the type of authentication that is being used, such as MD5, HMAC-SHA-1, and others. |
| `<name> key <key-id> plain-text <text>` | Specifies the password in plain text format. |
| `<name> key <key-id> encrypted-text <text>` | Specifies the password in an encrypted text format. |
| `<name> key prefer-key-id <key-id>` | Preferred key-id configuration will be used while sending out the packet with the specified key. |

- When an authentication profile is available, you can configure an authentication attachment point at the area or interface level.
- When an authentication profile contains multiple key-IDs, and the preferred key-ID is not configured, the packet is sent using the highest key-ID.

The authentication profile "auth-profile1" has `md5`, `hmac-sha-1`, and `clear-text` enabled. The preferred key-id being 20, the `hmac-sha-1` method will be used for authentication.

In the following configuration, the authentication profile is defined for each OSPF interface within a specific area. It specifies the authentication profile 'AUTH_PROFILE_SHA_512_1' for OSPF packet authentication and is applied to the interfaces 'ifl-0/0/0/1' and 'ifl-0/0/0/100', ensuring that OSPF neighbors on these interfaces use the same authentication mechanism to securely establish and maintain adjacencies.

This configuration also enables segment routing for this OSPF instance and defines route redistribution into OSPF. It specifies the source protocols (BGP and direct) whose routes will be imported into OSPF.

```
    "rtbrick-config:instance": [
      {
        "name": "default",
        "address-family": [
          {
            "afi": "ipv4",
            "safi": "unicast"
          },
          {
            "afi": "ipv6",
            "safi": "unicast"
          }
        ],
        "protocol": {
          "ospf": {
            "address-family": {
              "ipv6": {
                "instance-id": [
                  {
                    "instance-id": 0,
                    "router-id": "192.168.0.10",
                    "max-load-balance": 100,
                    "metric": 1000,
                    "segment-routing": {
                      "status": "enable",
                      "srgb": {
                        "base": 1000,
                        "range": 1000
                      }
                    },
                    "redistribute": [
                      {
                        "source": "bgp",
                        "metric": 2000
                      },
                      {
                        "source": "direct",
                        "policy": "ospf_policy_1"
                      }
                    ],
                    "area": [
                      {
                        "area-id": "0.0.0.0",
                        "metric": 1000,
                        "interface": [
                          {
```

```
                              "name": "ifl-0/0/0/1",
                              "authentication-profile": "AUTH_PROFILE_SHA_512_1"
                            },
                            {
                              "name": "ifl-0/0/0/100",
                              "metric": 20000,
                              "network-type": "p2p",
                              "authentication-profile": "AUTH_PROFILE_SHA_512_1"
```

API To access the RESTCONF API that corresponds to this CLI, click here.

## 2.7.3. OSPF Operational Commands

### OSPF Show Commands

#### OSPF Summary

Displays the OSPF protocol summary information.

**Syntax:**

**show ospf summary** <options>

| Option | Description |
|---|---|
| - | Without any option, the command displays the information for all instances and address families. |
| instance <instance-name> | OSPF summary information for the given instance. |
| instance <instance-name> <afi> | OSPF summary information for the specified instance and address family. Supported AFI values are 'ipv4' and 'ipv6'. |
| <afi> instance <instance-name> | OSPF summary information for the specified address family and instance. Supported AFI values are 'ipv4' and 'ipv6'. |

Example: OSPF summary for the default instance

```
supervisor@rtbrick: op> show ospf summary
Global Information:
  Neighbor State Information:
    Full       : 2
    Loading    : 0
    Exchange   : 0
    ExStart    : 0
    TwoWay     : 0
    Init       : 0
    Attempt    : 0
    Down       : 0
Instance: default, Address family: ipv4
```

```
  General information:
    Router ID: 198.51.100.11, Hostname: CBNG1, Area count: 2, Flood interval: 1000ms
    Opaque capability: True, Segment routing capability: True
    SR parameters:
      SRGB base: 10000, SRGB range: 1000, SRGB label values: 10000-10999
    Flags: OPAQUE|SR|-|-|-|ASBR|ABR, Cost: 4294967295
    SPF initial delay: 50ms, SPF short delay: 200ms, SPF long delay: 5000ms
  Area: 0.0.0.0
    Interface count: 1
      Interface: lo-0/0/0/0
        Address: 198.51.100.11, State: P2P, Type: p2p, Priority: 1
        Designated router: 0.0.0.0, Backup designated router: 0.0.0.0
        Hello interval: 10s, Dead interval: 40s
        Cost: 10000, MTU:
  Area: 0.0.0.1
    Interface count: 2
      Interface: ifl-0/0/2/0
        Address: 192.0.2.1, State: P2P, Type: p2p, Priority: 1
        Designated router: 0.0.0.0, Backup designated router: 0.0.0.0
        Hello interval: 10s, Dead interval: 40s
        Cost: 10000, MTU: 1500
      Interface: lo-0/0/0/1
        Address: 198.51.100.12, State: P2P, Type: p2p, Priority: 1
        Designated router: 0.0.0.0, Backup designated router: 0.0.0.0
        Hello interval: 10s, Dead interval: 40s
        Cost: 10000, MTU:
Instance: default, Address family: ipv6, Instance ID: 0
  General information:
    Router ID: 198.51.100.11, Hostname: CBNG1, Area count: 2, Flood interval: 1000ms
    Opaque capability: False, Segment routing capability: True
    SR parameters:
      SRGB base: 10000, SRGB range: 1000, SRGB label values: 10000-10999
    Flags: SR|-|-|-|-|ABR, Cost: 4294967295
    SPF initial delay: 50ms, SPF short delay: 200ms, SPF long delay: 5000ms
  Area: 0.0.0.0
    Interface count: 2
      Interface: ifl-0/0/2/100
        Address: fe80::9819:2c00:6406:b503, State: BDR, Type: broadcast, Priority: 1
        Designated router: 198.51.100.21, Backup designated router: 198.51.100.11
        Hello interval: 10s, Dead interval: 40s
        Cost: 10000, MTU: 1500
      Interface: lo-0/0/0/0
        Address: 2001:db8:0:1::11, State: P2P, Type: p2p, Priority: 1
        Designated router: 0.0.0.0, Backup designated router: 0.0.0.0
        Hello interval: 10s, Dead interval: 40s
        Cost: 10000, MTU: 0
  Area: 0.0.0.1
    Interface count: 2
      Interface: ifl-0/0/2/0
        Address: fe80::9819:2cff:fe06:b503, State: DR, Type: broadcast, Priority: 1
        Designated router: 198.51.100.11, Backup designated router: 0.0.0.0
```

```
        Hello interval: 10s, Dead interval: 40s
        Cost: 10000, MTU: 1500
      Interface: lo-0/0/0/1
        Address: 2001:db8:0:1::12, State: P2P, Type: p2p, Priority: 1
        Designated router: 0.0.0.0, Backup designated router: 0.0.0.0
        Hello interval: 10s, Dead interval: 40s
        Cost: 10000, MTU: 0
```

API To access the Operational State API that corresponds to this CLI, click include::common:partial$operational_api_links.adoc[tag=show_ospf_summary].

#### OSPF Hostname

Displays the OSPF hostname information.

**Syntax:**

**show ospf hostname**

Example: OSPF hostname for the default instance

```
supervisor@rtbrick: op> show ospf hostname
Instance                      AFI   Instance ID     Router ID                                   Hostname
default                       ipv4                  198.51.100.11                               CBNG1
default                       ipv4                  198.51.100.21                               CBNG2
default                       ipv6  0               198.51.100.11                               CBNG1
default                       ipv6  0               198.51.100.21                               CBNG2
```

#### OSPF Interface

Displays OSPF interface information.

**Syntax:**

**show ospf interface** <options>

| Option | Description |
|---|---|
| - | Without any option, the command displays the interface information for all instances and address families. |
| detail | Displays the detailed interface information. |
| <interface-name> detail | Displays detailed information for the specified interface. Also, for the specified interface, you can display interface information with filter options: detail, ipv4, and ipv4 detail. |

| Option | Description |
|---|---|
| `instance <instance-name>` | OSPF interface information for the given instance. Also, for the specified instance, you can display interface information with filter options: interface name, detail, and ipv4 detail. |
| `instance <instance-name> <interface-name>` | Displays information for a specified interface for a given instance. |
| `instance <instance-name> <interface-name> detail` | Displays detailed information for a specified interface for a given instance. |
| `<afi>` | Displays OSPF interface information for the specified address family. Supported AFI values are 'ipv4' and 'ipv6'. You can display interface information for the specified address family using filters such as interface name, detail, and instance. |

Example 1: OSPF interface information for the default instance

```
supervisor@rtbrick: op> show ospf interface
Instance: default, Address family: ipv4
  Interface            Area            IP Address                    State          Type       Cost     Priority DR
BDR             MTU
  ifl-0/0/2/0          0.0.0.1         192.0.2.1                     P2P            p2p        10000    1        0.0.0.0
0.0.0.0         1500
  lo-0/0/0/0           0.0.0.0         198.51.100.11                 P2P            p2p        10000    1        0.0.0.0
0.0.0.0
  lo-0/0/0/1           0.0.0.1         198.51.100.12                 P2P            p2p        10000    1        0.0.0.0
0.0.0.0
Instance: default, Address family: ipv6, Instance ID: 0
  Interface            Area            IP Address                    State          Type       Cost     Priority DR
BDR             MTU
  ifl-0/0/2/0          0.0.0.1         fe80::9819:2cff:fe06:b503     DR             broadcast  10000    1
198.51.100.11   0.0.0.0         1500
  ifl-0/0/2/100        0.0.0.0         fe80::9819:2c00:6406:b503     BDR            broadcast  10000    1
198.51.100.21   198.51.100.11   1500
  lo-0/0/0/0           0.0.0.0         2001:db8:0:1::11              P2P            p2p        10000    1        0.0.0.0
0.0.0.0         0
  lo-0/0/0/1           0.0.0.1         2001:db8:0:1::12              P2P            p2p        10000    1        0.0.0.0
0.0.0.0         0
```

Example 2: OSPF interface information

```
supervisor@rtbrick: op> show ospf interface ifl-0/0/2/0
Instance: default, Address family: ipv4
  Interface            Area            IP Address                    State          Type       Cost     Priority DR
BDR             MTU
  ifl-0/0/2/0          0.0.0.1         192.0.2.1                     P2P            p2p        10000    1        0.0.0.0
0.0.0.0         1500
Instance: default, Address family: ipv6, Instance ID: 0
  Interface            Area            IP Address                    State          Type       Cost     Priority DR
BDR             MTU
  ifl-0/0/2/0          0.0.0.1         fe80::9819:2cff:fe06:b503     DR             broadcast  10000    1
198.51.100.11   0.0.0.0         1500
```

API To access the Operational State API that corresponds to this CLI, click include::common:partial$operational_api_links.adoc[tag=show_ospf_interface].

#### OSPF Neighbor

Displays OSPF neighbor information.

**Syntax:**

**show ospf neighbor** <options>

| Option | Description |
|---|---|
| - | Without any option, the command displays the neighbor information for all instances. |
| instance <instance-name> | OSPF neighbor information for the given instance. |
| area <area-id> | OSPF neighbor information for the given area. |
| detail | Displays the detailed neighbor information. |
| interface <interface-name> | Displays the neighbor information for a specified interface. |
| instance <instance-name> detail | Displays detailed OSPF neighbor information for the given instance. |
| instance <instance-name> interface <interface-name> | Displays OSPF neighbor information for the specified interface for the given instance. |
| instance <instance-name> interface <interface-name> detail | Displays detailed OSPF neighbor information for the specified interface for the given instance. |
| interface <interface-name> detail | Displays detailed neighbor information for a specified interface. |
| <afi> | Displays OSPF neighbor information for the specified address family. Supported AFI values are 'ipv4' and 'ipv6'. You can display neighbor information for the specified address family using filters such as interface name, detail, and instance. |
| log | Logs neighbor event information. |

Example: OSPF neighbor information for the default instance

```
supervisor@rtbrick: op> show ospf neighbor
Instance: default, Address family: ipv4
  Address                       Interface           Router          Area            State      Priority  DR
BDR             Uptime          Expires
  192.0.2.2                      ifl-0/0/2/0         CBNG2           0.0.0.1         Full       1         0.0.0.0
0.0.0.0         0d:00h:00m:54s  36s
Instance: default, Address family: ipv6, Instance ID: 0
  Address                       Interface           Router          Area            State      Priority  DR
BDR             Uptime          Expires
  fe80::9819:2c00:6499:472d      ifl-0/0/2/100       CBNG2           0.0.0.0         Full       1         198.51.100.21
198.51.100.11   0d:00h:00m:10s  35s
```

API To access the Operational State API that corresponds to this CLI, click include::common:partial$operational_api_links.adoc[tag=show_ospf_neighbor].

#### OSPF Segment Routing

##### OSPF Segment Routing Prefix Segment

Displays OSPF segment routing prefix segment information.

**Syntax:**

**show ospf segment-routing prefix-segment** <options>

| Option | Description |
|---|---|
| - | Without any option, the command displays the neighbor information for all instances. |
| prefix-segment | Displays the OSPF prefix segment information |
| prefix-segment <instance> | Displays the OSPF prefix segments for the specified instance and address family |
| prefix-segment <instance-name> <afi> | Displays Segment routing global block (SRGB) information for the specified instance address family. Supported AFI values are 'ipv4' and 'ipv6'. |

Example: OSPF neighbor information for the default instance

```
supervisor@rtbrick: op> show ospf segment-routing prefix-segment
Instance: default, Address family: ipv4, Area ID: 0.0.0.0
  Advertising Router        Prefix                                    SID      Flags
  CBNG1                     198.51.100.11/32                          100          0
Instance: default, Address family: ipv4, Area ID: 0.0.0.1
  Advertising Router        Prefix                                    SID      Flags
  CBNG1                     198.51.100.11/32                          100          0
  CBNG2                     198.51.100.21/32                          200          0
Instance: default, Address family: ipv6, Instance ID: 0, Area ID: 0.0.0.0
  Advertising Router        Prefix                                    SID      Flags
  CBNG1                     2001:db8:0:1::11/128                      101          0
  CBNG2                     2001:db8:0:1::21/128                      201          0
Instance: default, Address family: ipv6, Instance ID: 0, Area ID: 0.0.0.1
  Advertising Router        Prefix                                    SID      Flags
  CBNG1                     2001:db8:0:1::11/128                      101          0
  CBNG1                     2001:db8:0:1::21/128                      201         64
```

##### OSPF Segment Routing Global Block

Displays OSPF segment routing global block (SRGB) information.

**Syntax:**

**show ospf segment-routing global-block** <options>

| Option | Description |
|---|---|
| - | Without any option, the command displays the neighbor information for all instances. |
| global-block | Displays Segment routing global block (SRGB) information |
| global-block instance <instance-name> | Displays Segment routing global block (SRGB) information for the specified instance |
| global-block instance <instance-name> <afi> | Displays Segment routing global block (SRGB) information for the specified instance address family. Supported AFI values are 'ipv4' and 'ipv6'. |

Example: OSPF neighbor information for the default instance

```
supervisor@rtbrick: op> show ospf segment-routing global-block
Instance: default, Address family: ipv4, Area ID: 0.0.0.0
  Advertising Router        SRGB Base   SRGB Range
  CBNG1                         10000         1000
Instance: default, Address family: ipv4, Area ID: 0.0.0.1
  Advertising Router        SRGB Base   SRGB Range
  CBNG1                         10000         1000
  CBNG2                         11000         1000
Instance: default, Address family: ipv6, Instance ID: 0, Area ID: 0.0.0.0
  Advertising Router        SRGB Base   SRGB Range
  CBNG1                         10000         1000
  CBNG2                         11000         1000
Instance: default, Address family: ipv6, Instance ID: 0, Area ID: 0.0.0.1
  Advertising Router        SRGB Base   SRGB Range
  CBNG1                         10000         1000
```

#### OSPF Database

Displays information from OSPF link-state database that contains data about link-state advertisements (LSAs).

**Syntax:**

**show ospf database** <options>

| Option | Description |
|---|---|
| advertising-router <router-id> | Displays LSDB information for the specified advertising router. |
| advertising-router <router-id> detail | Displays the detailed LSDB information for the specified advertising router. |
| advertising-router <router-id> ls-id <ls-id> | Displays the LSDB information for the specified link-state ID for the advertising router. |

| Option | Description |
|---|---|
| advertising-router <router-id> ls-type external | Displays the LSDB information for the specified LSA type for the advertising router. Link-state advertisement types include external, network, router, and summary. |
| detail | Displays detailed information from LSDB. |
| instance <instance-name> | Displays OSPF database information for the given instance. |
| ls-id <ls-id> | OSPF database information for a specific link-state identifier. |
| ls-type <type> | OSPF database information for the specified link-state type. Link-state advertisement types include external, network, router and summary. |
| area <area-id> | Displays database information for the specified OSPF area. |
| area <area-id> advertising-router <router-id> | Displays LSDB information for the specified advertising router for a specified OSPF area. |
| area <area-id> detail | Displays detailed LSDB information for the specified OSPF area. |
| area <area-id> ls-id <ls-id> | Displays LSDB information for the specified link-state identifier for the specified OSPF area. |
| area <area-id> ls-type <type> | Displays LSDB information for the specified link-state type for the specified OSPF area. Link-state advertisement types include external, network, router, and summary. |
| instance <instance-name> advertising router <router-id> | Displays LSDB information for the specified advertising router for the given instance. |
| instance <instance-name> area <area-id> | Displays LSDB information for the specified area for the given instance. |
| instance <instance-name> ls-id <ls-id> | Displays LSDB information for the specified link-state identifier for the given instance. |
| instance <instance-name> ls-type <type> | Displays LSDB information for the specified type of the given instance. Link-state advertisement types include external, network, router and summary. |
| <afi> | Displays LSDB neighbor information for the specified address family. Supported AFI values are 'ipv4' and 'ipv6'. You can display LSDB information for the specified address family using filters such as interface name, detail, and instance. |

Example 1: OSPF database information for the default instance

```
supervisor@rtbrick: op> show ospf database
Instance: default, Address family: ipv4, Area: 0.0.0.0
  Type                      Link State ID   Advertising Router          Age        Sequence   Checksum     Cost
  Router                    198.51.100.11   CBNG1                        62       0x80000001     0xf4a7        -
  Summary-Network           192.0.2.0       CBNG1                        61       0x80000001     0x3d81    10000
  Summary-Network           198.51.100.12   CBNG1                        61       0x80000001     0xe72c    10000
  Summary-Network           198.51.100.22   CBNG1                        53       0x80000001     0x844e    20000
  Summary-Asbr              198.51.100.21   CBNG1                        53       0x80000001     0x7f8a    10000
```

```
  Opaque-Area              4.0.0.0         CBNG1                        62      0x80000001     0x1c15        -
  Opaque-Area              7.0.0.100       CBNG1                        62      0x80000001     0xf62e        -
Instance: default, Address family: ipv4, Area: 0.0.0.1
  Type                     Link State ID   Advertising Router          Age        Sequence   Checksum     Cost
  Router                   198.51.100.11   CBNG1                        58      0x80000002     0x1304        -
  Router                   198.51.100.21   CBNG2                        59      0x80000002     0x7190        -
  Summary-Network          192.0.2.4       CBNG2                        59      0x80000001     0xd8d7    10000
  Summary-Network          198.51.100.11   CBNG1                        61      0x80000001     0xf123    10000
  Summary-Network          198.51.100.21   CBNG2                        59      0x80000001     0x51af    10000
  Opaque-Area              4.0.0.0         CBNG1                        62      0x80000001     0x1c15        -
  Opaque-Area              4.0.0.0         CBNG2                        65      0x80000001      0xa30        -
  Opaque-Area              7.0.0.100       CBNG1                        61      0x80000001      0xa98        -
  Opaque-Area              7.0.0.200       CBNG2                        59      0x80000001     0xa025        -
Instance: default, Address family: ipv4
  Type                     Link State ID   Advertising Router          Age        Sequence   Checksum     Cost
  External                 192.0.2.0       CBNG1                        62      0x80000001     0x26c6 16777214
  External                 192.0.2.0       CBNG2                        65      0x80000001     0xe9f8 16777214
  External                 192.0.2.4       CBNG1                        62      0x80000001     0xfdea 16777214
  External                 192.0.2.4       CBNG2                        65      0x80000001     0xc11d 16777214
  External                 198.51.100.11   CBNG1                        62      0x80000001     0xda68 16777214
  External                 198.51.100.12   CBNG1                        62      0x80000001     0xd071 16777214
  External                 198.51.100.21   CBNG2                        65      0x80000001     0x3af4 16777214
  External                 198.51.100.22   CBNG2                        65      0x80000001     0x30fd 16777214
Instance: default, Address family: ipv6, Instance ID: 0, Area: 0.0.0.0
  Type                     Link State ID   Advertising Router          Age        Sequence   Checksum     Cost
  OSPFv3-Router            0.0.0.0         CBNG1                        14      0x80000002     0xe3dc        -
  OSPFv3-Router            0.0.0.0         CBNG2                        15      0x80000002     0xacf5        -
  OSPFv3-Network           1.28.144.1      CBNG2                        15      0x80000001     0x79b0        -
  Inter-Area-Prefix        1.0.0.0         CBNG1                        62      0x80000001     0x4f1d    10000
  Inter-Area-Prefix        1.0.0.0         CBNG2                        59      0x80000001     0x5c9a    10000
  Inter-Area-Prefix        2.0.0.0         CBNG1                        62      0x80000001     0xec23    10000
  Intra-Area-Prefix        1.0.0.0         CBNG1                        14      0x80000004     0x5421        -
  Intra-Area-Prefix        1.0.0.0         CBNG2                        15      0x80000003     0xc889        -
  Intra-Area-Prefix        118.32.3.0      CBNG2                        15      0x80000001     0x6448        -
  OSPFv3-Router-Information 0.0.0.0        CBNG1                        62      0x80000001     0x4349        -
  OSPFv3-Router-Information 0.0.0.0        CBNG2                        65      0x80000001     0x3164        -
  E-Intra-Area-Prefix      1.0.0.0         CBNG1                        62      0x80000001     0x7b8c        -
  E-Intra-Area-Prefix      1.0.0.0         CBNG2                        60      0x80000001     0x9be3        -
  Link                     1.8.144.1       CBNG1                        19      0x80000003     0x2e4a        -
  Link                     1.28.144.1      CBNG2                        20      0x80000002     0x2fdc        -
Instance: default, Address family: ipv6, Instance ID: 0, Area: 0.0.0.1
  Type                     Link State ID   Advertising Router          Age        Sequence   Checksum     Cost
  OSPFv3-Router            0.0.0.0         CBNG1                        62      0x80000001     0x9632        -
  Inter-Area-Prefix        1.0.0.0         CBNG1                        62      0x80000001     0x5f0c    10000
  Inter-Area-Prefix        2.0.0.0         CBNG1                        62      0x80000001     0xd23e    10000
  Inter-Area-Prefix        3.0.0.0         CBNG1                         8      0x80000001     0x9831    20000
  Inter-Area-Prefix        4.0.0.0         CBNG1                         8      0x80000001     0x7455    20000
  Intra-Area-Prefix        1.0.0.0         CBNG1                        22      0x80000002     0x1cec        -
  OSPFv3-Router-Information 0.0.0.0        CBNG1                        62      0x80000001     0x4349        -
  E-Inter-Area-Prefix      1.0.0.0         CBNG1                        62      0x80000001     0xdc06        -
  E-Inter-Area-Prefix      2.0.0.0         CBNG1                         8      0x80000001     0xde19        -
  Link                     1.8.0.0         CBNG1                        22      0x80000002     0x5918        -
```

Example 2: OSPF database detailed information

```
supervisor@rtbrick: op> show ospf database detail
Instance: default, Address family: ipv4, Area: 0.0.0.0 LSAs
  LSA ID: 198.51.100.11
    Advertising router: CBNG1, LSA type: Router
    Sequence number: 0x80000001, Checksum: 0xf4a7, LSA age: 63s
```

```
    Length: 36, Options: *|O|-|-|-|-|E|*, Flags: -|-|E|B
    Number of links: 1
      Link ID: 198.51.100.11
        Link data: 255.255.255.255, Type: Stub
        Type of service: 0, Metric: 10000
  LSA ID: 192.0.2.0
    Advertising router: CBNG1, LSA type: Summary-Network
    Sequence number: 0x80000001, Checksum: 0x3d81, LSA age: 62s
    Length: 28, Options: *|O|-|-|-|-|-|*
    Mask: 255.255.255.252, Metric: 10000
  LSA ID: 198.51.100.12
    Advertising router: CBNG1, LSA type: Summary-Network
    Sequence number: 0x80000001, Checksum: 0xe72c, LSA age: 62s
    Length: 28, Options: *|O|-|-|-|-|-|*
    Mask: 255.255.255.255, Metric: 10000
  LSA ID: 198.51.100.22
    Advertising router: CBNG1, LSA type: Summary-Network
    Sequence number: 0x80000001, Checksum: 0x844e, LSA age: 54s
    Length: 28, Options: *|O|-|-|-|-|-|*
    Mask: 255.255.255.255, Metric: 20000
  LSA ID: 198.51.100.21
    Advertising router: CBNG1, LSA type: Summary-Asbr
    Sequence number: 0x80000001, Checksum: 0x7f8a, LSA age: 54s
    Length: 28, Options: *|O|-|-|-|-|-|*
    Mask: 0.0.0.0, Metric: 10000
  LSA ID: 4.0.0.0
    Advertising router: CBNG1, LSA type: Opaque-Area
    Sequence number: 0x80000001, Checksum: 0x1c15, LSA age: 63s
    Length: 56, Options: *|-|-|-|-|-|-|*
    Hostname mapping:
      Hostname: CBNG1, Router ID: 198.51.100.11
    Segment routing range TLV:
      Range size: 1000, Label: 10000
  LSA ID: 7.0.0.100
    Advertising router: CBNG1, LSA type: Opaque-Area
    Sequence number: 0x80000001, Checksum: 0xf62e, LSA age: 63s
    Length: 44, Options: *|-|-|-|-|-|-|*
    Extended prefix TLV:
      Prefix: 198.51.100.11/32
      Route type: 1, AF: 0, Flags: 64
      Prefix SID sub TLV:
        Multi topology ID: 0, Algorithm: 0, Flags: 0, SID: 100
Instance: default, Address family: ipv4, Area: 0.0.0.1 LSAs
  LSA ID: 198.51.100.11
    Advertising router: CBNG1, LSA type: Router
    Sequence number: 0x80000002, Checksum: 0x1304, LSA age: 59s
    Length: 60, Options: *|O|-|-|-|-|E|*, Flags: -|-|E|B
    Number of links: 3
      Link ID: 198.51.100.21
        Link data: 192.0.2.1, Type: P2P
        Type of service: 0, Metric: 10000
      Link ID: 192.0.2.0
        Link data: 255.255.255.252, Type: Stub
        Type of service: 0, Metric: 10000
      Link ID: 198.51.100.12
        Link data: 255.255.255.255, Type: Stub
        Type of service: 0, Metric: 10000
  LSA ID: 198.51.100.21
```

```
    Advertising router: CBNG2, LSA type: Router, Router ID: CBNG2
    Sequence number: 0x80000002, Checksum: 0x7190, LSA age: 60s
    Interface: ifl-0/0/2/0, Neighbor address: 192.0.2.2
    Length: 60, Options: *|O|-|-|-|-|E|*, Flags: -|-|E|B
    Number of links: 3
      Link ID: 198.51.100.11
        Link data: 192.0.2.2, Type: P2P
        Type of service: 0, Metric: 10000
      Link ID: 192.0.2.0
        Link data: 255.255.255.252, Type: Stub
        Type of service: 0, Metric: 10000
      Link ID: 198.51.100.22
        Link data: 255.255.255.255, Type: Stub
        Type of service: 0, Metric: 10000
  LSA ID: 192.0.2.4
    Advertising router: CBNG2, LSA type: Summary-Network, Router ID: CBNG2
    Sequence number: 0x80000001, Checksum: 0xd8d7, LSA age: 60s
    Interface: ifl-0/0/2/0, Neighbor address: 192.0.2.2
    Length: 28, Options: *|O|-|-|-|-|-|*
    Mask: 255.255.255.252, Metric: 10000
  LSA ID: 198.51.100.11
    Advertising router: CBNG1, LSA type: Summary-Network
    Sequence number: 0x80000001, Checksum: 0xf123, LSA age: 62s
    Length: 28, Options: *|O|-|-|-|-|-|*
    Mask: 255.255.255.255, Metric: 10000
  LSA ID: 198.51.100.21
    Advertising router: CBNG2, LSA type: Summary-Network, Router ID: CBNG2
    Sequence number: 0x80000001, Checksum: 0x51af, LSA age: 60s
    Interface: ifl-0/0/2/0, Neighbor address: 192.0.2.2
    Length: 28, Options: *|O|-|-|-|-|-|*
    Mask: 255.255.255.255, Metric: 10000
  LSA ID: 4.0.0.0
    Advertising router: CBNG1, LSA type: Opaque-Area
    Sequence number: 0x80000001, Checksum: 0x1c15, LSA age: 63s
    Length: 56, Options: *|-|-|-|-|-|-|*
    Hostname mapping:
      Hostname: CBNG1, Router ID: 198.51.100.11
    Segment routing range TLV:
      Range size: 1000, Label: 10000
  LSA ID: 4.0.0.0
    Advertising router: CBNG2, LSA type: Opaque-Area, Router ID: CBNG2
    Sequence number: 0x80000001, Checksum: 0xa30, LSA age: 66s
    Interface: ifl-0/0/2/0, Neighbor address: 192.0.2.2
    Length: 56, Options: *|-|-|-|-|-|-|*
    Hostname mapping:
      Hostname: CBNG2, Router ID: 198.51.100.21
    Segment routing range TLV:
      Range size: 1000, Label: 11000
  LSA ID: 7.0.0.100
    Advertising router: CBNG1, LSA type: Opaque-Area
    Sequence number: 0x80000001, Checksum: 0xa98, LSA age: 62s
    Length: 44, Options: *|-|-|-|-|-|-|*
    Extended prefix TLV:
      Prefix: 198.51.100.11/32
      Route type: 3, AF: 0, Flags: 192
      Prefix SID sub TLV:
        Multi topology ID: 0, Algorithm: 0, Flags: 0, SID: 100
  LSA ID: 7.0.0.200
```

```
    Advertising router: CBNG2, LSA type: Opaque-Area, Router ID: CBNG2
    Sequence number: 0x80000001, Checksum: 0xa025, LSA age: 60s
    Interface: ifl-0/0/2/0, Neighbor address: 192.0.2.2
    Length: 44, Options: *|-|-|-|-|-|-|*
    Extended prefix TLV:
      Prefix: 198.51.100.21/32
      Route type: 3, AF: 0, Flags: 192
      Prefix SID sub TLV:
        Multi topology ID: 0, Algorithm: 0
        Reserved: 0, Flags: 0, SID: 200
Instance: default, Address family: ipv4 LSAs
  LSA ID: 192.0.2.0
    Advertising router: CBNG1, LSA type: External
    Sequence number: 0x80000001, Checksum: 0x26c6, LSA age: 63s
    Length: 36, Options: *|O|-|-|-|-|-|*
    Forwarding address: 0.0.0.0, Mask: 255.255.255.252, Route tag: 0
    Metric: 16777214, Metric type: EXTERNAL_TYPE_1
  LSA ID: 192.0.2.0
    Advertising router: CBNG2, LSA type: External, Router ID: CBNG2
    Sequence number: 0x80000001, Checksum: 0xe9f8, LSA age: 66s
    Interface: ifl-0/0/2/0, Neighbor address: 192.0.2.2
    Length: 36, Options: *|O|-|-|-|-|-|*
    Forwarding address: 0.0.0.0, Mask: 255.255.255.252, Route tag: 0
    Metric: 16777214, Metric type: EXTERNAL_TYPE_1
  LSA ID: 192.0.2.4
    Advertising router: CBNG1, LSA type: External
    Sequence number: 0x80000001, Checksum: 0xfdea, LSA age: 63s
    Length: 36, Options: *|O|-|-|-|-|-|*
    Forwarding address: 0.0.0.0, Mask: 255.255.255.252, Route tag: 0
    Metric: 16777214, Metric type: EXTERNAL_TYPE_1
  LSA ID: 192.0.2.4
    Advertising router: CBNG2, LSA type: External, Router ID: CBNG2
    Sequence number: 0x80000001, Checksum: 0xc11d, LSA age: 66s
    Interface: ifl-0/0/2/0, Neighbor address: 192.0.2.2
    Length: 36, Options: *|O|-|-|-|-|-|*
    Forwarding address: 0.0.0.0, Mask: 255.255.255.252, Route tag: 0
    Metric: 16777214, Metric type: EXTERNAL_TYPE_1
  LSA ID: 198.51.100.11
    Advertising router: CBNG1, LSA type: External
    Sequence number: 0x80000001, Checksum: 0xda68, LSA age: 63s
    Length: 36, Options: *|O|-|-|-|-|-|*
    Forwarding address: 0.0.0.0, Mask: 255.255.255.255, Route tag: 0
    Metric: 16777214, Metric type: EXTERNAL_TYPE_1
  LSA ID: 198.51.100.12
    Advertising router: CBNG1, LSA type: External
    Sequence number: 0x80000001, Checksum: 0xd071, LSA age: 63s
    Length: 36, Options: *|O|-|-|-|-|-|*
    Forwarding address: 0.0.0.0, Mask: 255.255.255.255, Route tag: 0
    Metric: 16777214, Metric type: EXTERNAL_TYPE_1
  LSA ID: 198.51.100.21
    Advertising router: CBNG2, LSA type: External, Router ID: CBNG2
    Sequence number: 0x80000001, Checksum: 0x3af4, LSA age: 66s
    Interface: ifl-0/0/2/0, Neighbor address: 192.0.2.2
    Length: 36, Options: *|O|-|-|-|-|-|*
    Forwarding address: 0.0.0.0, Mask: 255.255.255.255, Route tag: 0
    Metric: 16777214, Metric type: EXTERNAL_TYPE_1
  LSA ID: 198.51.100.22
    Advertising router: CBNG2, LSA type: External, Router ID: CBNG2
```

```
    Sequence number: 0x80000001, Checksum: 0x30fd, LSA age: 66s
    Interface: ifl-0/0/2/0, Neighbor address: 192.0.2.2
    Length: 36, Options: *|O|-|-|-|-|-|*
    Forwarding address: 0.0.0.0, Mask: 255.255.255.255, Route tag: 0
    Metric: 16777214, Metric type: EXTERNAL_TYPE_1
Instance: default, Address family: ipv6, Area: 0.0.0.0, Instance ID: 0 LSAs
  LSA ID: 0.0.0.0
    Advertising router: CBNG1, LSA type: OSPFv3-Router
    Sequence number: 0x80000002, Checksum: 0xe3dc, LSA age: 15s
    Length: 40, Options: -|V6|E|-|-|R|-|-, Flags: -|-|-|B
    Number of links: 1
    Link Entry: 1
      Type: Transit, Metric: 10000
      Local interface ID: 26216449, Neighbor interface ID: 26221569, Neighbor router ID: 198.51.100.21
  LSA ID: 0.0.0.0
    Advertising router: CBNG2, LSA type: OSPFv3-Router, Router ID: CBNG2
    Sequence number: 0x80000002, Checksum: 0xacf5, LSA age: 16s
    Interface: ifl-0/0/2/100, Neighbor address: fe80::9819:2c00:6499:472d
    Length: 40, Options: -|V6|E|-|-|R|-|-, Flags: -|-|-|B
    Number of links: 1
    Link Entry: 1
      Type: Transit, Metric: 10000
      Local interface ID: 26221569, Neighbor interface ID: 26221569, Neighbor router ID: 198.51.100.21
  LSA ID: 1.28.144.1
    Advertising router: CBNG2, LSA type: OSPFv3-Network, Router ID: CBNG2
    Sequence number: 0x80000001, Checksum: 0x79b0, LSA age: 16s
    Interface: ifl-0/0/2/100, Neighbor address: fe80::9819:2c00:6499:472d
    Length: 32, Options: -|V6|E|-|-|R|-|-, Number of links: 2
    Attached routers: 198.51.100.11, 198.51.100.21
  LSA ID: 1.0.0.0
    Advertising router: CBNG1, LSA type: Inter-Area-Prefix
    Sequence number: 0x80000001, Checksum: 0x4f1d, LSA age: 63s
    Length: 36
    Metric: 10000, Prefix options: -|-|-|-|-|-, Prefix: 2001:db8:1:1::/64
  LSA ID: 1.0.0.0
    Advertising router: CBNG2, LSA type: Inter-Area-Prefix, Router ID: CBNG2
    Sequence number: 0x80000001, Checksum: 0x5c9a, LSA age: 60s
    Interface: ifl-0/0/2/100, Neighbor address: fe80::9819:2c00:6499:472d
    Length: 44
    Metric: 10000, Prefix options: -|-|LA|-|-|-, Prefix: 2001:db8:0:1::22/128
  LSA ID: 2.0.0.0
    Advertising router: CBNG1, LSA type: Inter-Area-Prefix
    Sequence number: 0x80000001, Checksum: 0xec23, LSA age: 63s
    Length: 44
    Metric: 10000, Prefix options: -|-|LA|-|-|-, Prefix: 2001:db8:0:1::12/128
  LSA ID: 1.0.0.0
    Advertising router: CBNG1, LSA type: Intra-Area-Prefix
    Sequence number: 0x80000004, Checksum: 0x5421, LSA age: 15s
    Length: 52
    Ref Type: OSPFv3-Router, Ref ID: 0.0.0.0, Ref Advertised Router: 198.51.100.11
    Prefix Count: 1
    Prefix List:
      Prefix: 2001:db8:0:1::11/128, Metric: 10000, Prefix options: -|-|LA|-|-|-
  LSA ID: 1.0.0.0
    Advertising router: CBNG2, LSA type: Intra-Area-Prefix, Router ID: CBNG2
    Sequence number: 0x80000003, Checksum: 0xc889, LSA age: 16s
    Interface: ifl-0/0/2/100, Neighbor address: fe80::9819:2c00:6499:472d
    Length: 52
```

```
    Ref Type: OSPFv3-Router, Ref ID: 0.0.0.0, Ref Advertised Router: 198.51.100.21
    Prefix Count: 1
    Prefix List:
      Prefix: 2001:db8:0:1::21/128, Metric: 10000, Prefix options: -|-|LA|-|-|-
  LSA ID: 118.32.3.0
    Advertising router: CBNG2, LSA type: Intra-Area-Prefix, Router ID: CBNG2
    Sequence number: 0x80000001, Checksum: 0x6448, LSA age: 16s
    Interface: ifl-0/0/2/100, Neighbor address: fe80::9819:2c00:6499:472d
    Length: 44
    Ref Type: OSPFv3-Network, Ref ID: 1.28.144.1, Ref Advertised Router: 198.51.100.21
    Prefix Count: 1
    Prefix List:
      Prefix: 2001:db8:2:1::/64, Metric: 0, Prefix options: -|-|-|-|-|-
  LSA ID: 0.0.0.0
    Advertising router: CBNG1, LSA type: OSPFv3-Router-Information
    Sequence number: 0x80000001, Checksum: 0x4349, LSA age: 63s
    Length: 56
    Hostname mapping:
      Hostname: CBNG1, Router ID: 198.51.100.11
    Segment routing range TLV:
      Range size: 1000, Label: 10000
  LSA ID: 0.0.0.0
    Advertising router: CBNG2, LSA type: OSPFv3-Router-Information, Router ID: CBNG2
    Sequence number: 0x80000001, Checksum: 0x3164, LSA age: 66s
    Interface: ifl-0/0/2/100, Neighbor address: fe80::9819:2c00:6499:472d
    Length: 56
    Hostname mapping:
      Hostname: CBNG2, Router ID: 198.51.100.21
    Segment routing range TLV:
      Range size: 1000, Label: 11000
  LSA ID: 1.0.0.0
    Advertising router: CBNG1, LSA type: E-Intra-Area-Prefix
    Sequence number: 0x80000001, Checksum: 0x7b8c, LSA age: 63s
    Length: 72
    Prefix Count: 1
    Prefix List:
      Prefix: 2001:db8:0:1::11/128, Metric: 10000, Prefix options: -|-|LA|-|-|-
        Extended prefix TLV:
        Algorithm: 0, Flags: 0, SID: 101
  LSA ID: 1.0.0.0
    Advertising router: CBNG2, LSA type: E-Intra-Area-Prefix, Router ID: CBNG2
    Sequence number: 0x80000001, Checksum: 0x9be3, LSA age: 61s
    Interface: ifl-0/0/2/100, Neighbor address: fe80::9819:2c00:6499:472d
    Length: 72
    Prefix Count: 1
    Prefix List:
      Prefix: 2001:db8:0:1::21/128, Metric: 10000, Prefix options: -|-|LA|-|-|-
        Extended prefix TLV:
        Algorithm: 0, Flags: 0, SID: 201
  LSA ID: 1.8.144.1
    Advertising router: CBNG1, LSA type: Link, Router ID: 0.0.0.0
    Sequence number: 0x80000003, Checksum: 0x2e4a, LSA age: 20s
    Interface: ifl-0/0/2/100
    Length: 56, Options: -|V6|E|-|-|R|-|-
    Prefix Count: 1
    Prefix List:
      Prefix: 2001:db8:2:1::/64, Metric: 0, Prefix options: -|-|-|-|-|-
  LSA ID: 1.28.144.1
```

```
    Advertising router: CBNG2, LSA type: Link, Router ID: CBNG2
    Sequence number: 0x80000002, Checksum: 0x2fdc, LSA age: 21s
    Interface: ifl-0/0/2/100, Neighbor address: fe80::9819:2c00:6499:472d
    Length: 56, Options: -|V6|E|-|-|R|-|-
    Prefix Count: 1
    Prefix List:
      Prefix: 2001:db8:2:1::/64, Metric: 0, Prefix options: -|-|-|-|-|-
Instance: default, Address family: ipv6, Area: 0.0.0.1, Instance ID: 0 LSAs
  LSA ID: 0.0.0.0
    Advertising router: CBNG1, LSA type: OSPFv3-Router
    Sequence number: 0x80000001, Checksum: 0x9632, LSA age: 63s
    Length: 24, Options: -|V6|E|-|-|R|-|-, Flags: -|-|-|-
    Number of links: 0
  LSA ID: 1.0.0.0
    Advertising router: CBNG1, LSA type: Inter-Area-Prefix
    Sequence number: 0x80000001, Checksum: 0x5f0c, LSA age: 63s
    Length: 36
    Metric: 10000, Prefix options: -|-|-|-|-|-, Prefix: 2001:db8:2:1::/64
  LSA ID: 2.0.0.0
    Advertising router: CBNG1, LSA type: Inter-Area-Prefix
    Sequence number: 0x80000001, Checksum: 0xd23e, LSA age: 63s
    Length: 44
    Metric: 10000, Prefix options: -|-|LA|-|-|-, Prefix: 2001:db8:0:1::11/128
  LSA ID: 3.0.0.0
    Advertising router: CBNG1, LSA type: Inter-Area-Prefix
    Sequence number: 0x80000001, Checksum: 0x9831, LSA age: 9s
    Length: 44
    Metric: 20000, Prefix options: -|-|-|-|-|-, Prefix: 2001:db8:0:1::22/128
  LSA ID: 4.0.0.0
    Advertising router: CBNG1, LSA type: Inter-Area-Prefix
    Sequence number: 0x80000001, Checksum: 0x7455, LSA age: 9s
    Length: 44
    Metric: 20000, Prefix options: -|-|-|-|-|-, Prefix: 2001:db8:0:1::21/128
  LSA ID: 1.0.0.0
    Advertising router: CBNG1, LSA type: Intra-Area-Prefix
    Sequence number: 0x80000002, Checksum: 0x1cec, LSA age: 23s
    Length: 64
    Ref Type: OSPFv3-Router, Ref ID: 0.0.0.0, Ref Advertised Router: 198.51.100.11
    Prefix Count: 2
    Prefix List:
      Prefix: 2001:db8:1:1::/64, Metric: 10000, Prefix options: -|-|-|-|-|-
      Prefix: 2001:db8:0:1::12/128, Metric: 10000, Prefix options: -|-|LA|-|-|-
  LSA ID: 0.0.0.0
    Advertising router: CBNG1, LSA type: OSPFv3-Router-Information
    Sequence number: 0x80000001, Checksum: 0x4349, LSA age: 63s
    Length: 56
    Hostname mapping:
      Hostname: CBNG1, Router ID: 198.51.100.11
    Segment routing range TLV:
      Range size: 1000, Label: 10000
  LSA ID: 1.0.0.0
    Advertising router: CBNG1, LSA type: E-Inter-Area-Prefix
    Sequence number: 0x80000001, Checksum: 0xdc06, LSA age: 63s
    Length: 60
    Prefix Count: 1
    Prefix List:
      Prefix: 2001:db8:0:1::11/128, Metric: 10000, Prefix options: -|-|LA|-|-|-
        Extended prefix TLV:
```

```
        Algorithm: 0, Flags: 0, SID: 101
  LSA ID: 2.0.0.0
    Advertising router: CBNG1, LSA type: E-Inter-Area-Prefix
    Sequence number: 0x80000001, Checksum: 0xde19, LSA age: 9s
    Length: 60
    Prefix Count: 1
    Prefix List:
      Prefix: 2001:db8:0:1::21/128, Metric: 20000, Prefix options: -|-|-|-|-|-
        Extended prefix TLV:
        Algorithm: 0, Flags: 64, SID: 201
  LSA ID: 1.8.0.0
    Advertising router: CBNG1, LSA type: Link, Router ID: 0.0.0.0
    Sequence number: 0x80000002, Checksum: 0x5918, LSA age: 23s
    Interface: ifl-0/0/2/0
    Length: 56, Options: -|V6|E|-|-|R|-|-
    Prefix Count: 1
    Prefix List:
      Prefix: 2001:db8:1:1::/64, Metric: 0, Prefix options: -|-|-|-|-|-
```

Example 3: OSPF database for an advertising router

```
supervisor@rtbrick: op> show ospf database advertising-router 198.51.100.11
Instance: default, Address family: ipv4, Area: 0.0.0.0
  Type                      Link State ID   Advertising Router          Age        Sequence   Checksum     Cost
  Router                    198.51.100.11   CBNG1                        65      0x80000001     0xf4a7        -
  Summary-Network           192.0.2.0       CBNG1                        64      0x80000001     0x3d81    10000
  Summary-Network           198.51.100.12   CBNG1                        64      0x80000001     0xe72c    10000
  Summary-Network           198.51.100.22   CBNG1                        56      0x80000001     0x844e    20000
  Summary-Asbr              198.51.100.21   CBNG1                        56      0x80000001     0x7f8a    10000
  Opaque-Area               4.0.0.0         CBNG1                        65      0x80000001     0x1c15        -
  Opaque-Area               7.0.0.100       CBNG1                        65      0x80000001     0xf62e        -
Instance: default, Address family: ipv4, Area: 0.0.0.1
  Type                      Link State ID   Advertising Router          Age        Sequence   Checksum     Cost
  Router                    198.51.100.11   CBNG1                        61      0x80000002     0x1304        -
  Summary-Network           198.51.100.11   CBNG1                        64      0x80000001     0xf123    10000
  Opaque-Area               4.0.0.0         CBNG1                        65      0x80000001     0x1c15        -
  Opaque-Area               7.0.0.100       CBNG1                        64      0x80000001      0xa98        -
Instance: default, Address family: ipv4
  Type                      Link State ID   Advertising Router          Age        Sequence   Checksum     Cost
  External                  192.0.2.0       CBNG1                        65      0x80000001     0x26c6 16777214
  External                  192.0.2.4       CBNG1                        65      0x80000001     0xfdea 16777214
  External                  198.51.100.11   CBNG1                        65      0x80000001     0xda68 16777214
  External                  198.51.100.12   CBNG1                        65      0x80000001     0xd071 16777214
Instance: default, Address family: ipv6, Instance ID: 0, Area: 0.0.0.0
  Type                      Link State ID   Advertising Router          Age        Sequence   Checksum     Cost
  OSPFv3-Router             0.0.0.0         CBNG1                        17      0x80000002     0xe3dc        -
  Inter-Area-Prefix         1.0.0.0         CBNG1                        65      0x80000001     0x4f1d    10000
  Inter-Area-Prefix         2.0.0.0         CBNG1                        65      0x80000001     0xec23    10000
  Intra-Area-Prefix         1.0.0.0         CBNG1                        17      0x80000004     0x5421        -
  OSPFv3-Router-Information 0.0.0.0         CBNG1                        65      0x80000001     0x4349        -
  E-Intra-Area-Prefix       1.0.0.0         CBNG1                        65      0x80000001     0x7b8c        -
  Link                      1.8.144.1       CBNG1                        22      0x80000003     0x2e4a        -
Instance: default, Address family: ipv6, Instance ID: 0, Area: 0.0.0.1
  Type                      Link State ID   Advertising Router          Age        Sequence   Checksum     Cost
  OSPFv3-Router             0.0.0.0         CBNG1                        65      0x80000001     0x9632        -
  Inter-Area-Prefix         1.0.0.0         CBNG1                        65      0x80000001     0x5f0c    10000
  Inter-Area-Prefix         2.0.0.0         CBNG1                        65      0x80000001     0xd23e    10000
  Inter-Area-Prefix         3.0.0.0         CBNG1                        11      0x80000001     0x9831    20000
  Inter-Area-Prefix         4.0.0.0         CBNG1                        11      0x80000001     0x7455    20000
```

```
  Intra-Area-Prefix         1.0.0.0         CBNG1                       25      0x80000002     0x1cec        -
  OSPFv3-Router-Information 0.0.0.0         CBNG1                       65      0x80000001     0x4349        -
  E-Inter-Area-Prefix       1.0.0.0         CBNG1                       65      0x80000001     0xdc06        -
  E-Inter-Area-Prefix       2.0.0.0         CBNG1                       11      0x80000001     0xde19        -
  Link                      1.8.0.0         CBNG1                       25      0x80000002     0x5918        -
```

#### OSPF SPF Result

Displays SPF results.

**Syntax:**

**show ospf spf result** <options>

| Option | Description |
|---|---|
| - | Without any option, the command displays the SPF result of all instances. |
| area <area-id> | Displays SPF result for the specified area. |
| instance <instance-name> | Name of the instance |
| node-id <node-id> | Displays SPF result for the specified node identifier. |
| area <area-id> <node-id> | Displays SPF result for the specified node identifier for a specified area. |
| instance <instance-name> area <area-id> | Displays SPF result for the specified area for a given instance. |
| instance <instance-name> node-id <node-id> | Displays SPF result for the specified node identifier for a given instance. |
| <afi> | Displays SPF result information for the specified address family. Supported AFI values are 'ipv4' and 'ipv6'. You can display SPF result information for the specified address family using filters such as interface name, detail, and instance. |

Example 1: OSPF SPF Result for the default instance

```
supervisor@rtbrick: op> show ospf spf result
Instance: default, Address family: ipv4, Area: 0.0.0.0
  Node ID         Type            Cost     Advertising Router  Flags        Neighbor Node         Interface
Nexthop
  198.51.100.11   router          0        CBNG1               -|-|E|B      -                     local            -
Instance: default, Address family: ipv4, Area: 0.0.0.1
  Node ID         Type            Cost     Advertising Router  Flags        Neighbor Node         Interface
Nexthop
  198.51.100.11   router          0        CBNG1               -|-|E|B      -                     local            -
  198.51.100.21   router          10000    CBNG2               -|-|E|B      198.51.100.21         ifl-0/0/2/0
192.0.2.2
Instance: default, Address family: ipv6, Area: 0.0.0.0, Instance ID: 0
  Node ID         Type            Cost     Advertising Router  Flags        Neighbor Node         Interface
Nexthop
  198.51.100.11   router          0        CBNG1               -|-|-|B      -                     local            -
  1.28.144.1      network         10000    CBNG2               -|-|-|-      -                     local            -
```

```
  198.51.100.21   router          10000    CBNG2              -|-|-|B      198.51.100.21         ifl-0/0/2/100
fe80::9819:2c00:6499:472d
Instance: default, Address family: ipv6, Area: 0.0.0.1, Instance ID: 0
  Node ID         Type            Cost     Advertising Router  Flags        Neighbor Node         Interface
Nexthop
  198.51.100.11   router          0        CBNG1              -|-|-|-      -                     local            -
```

Example 2: OSPF SPF Result for the specified node identifier for the given area

```
supervisor@rtbrick: op> show ospf spf result area 0.0.0.0 node-id 198.51.100.11
Instance: default, Address family: ipv4, Area: 0.0.0.0
  Node ID         Type            Cost     Advertising Router  Flags        Neighbor Node         Interface
Nexthop
  198.51.100.11   router          0        CBNG1              -|-|E|B      -                     local            -
Instance: default, Address family: ipv6, Area: 0.0.0.0, Instance ID: 0
  Node ID         Type            Cost     Advertising Router  Flags        Neighbor Node         Interface
Nexthop
  198.51.100.11   router          0        CBNG1              -|-|-|B      -                     local            -
```

#### OSPF SPF Log

Displays SPF Log information.

**Syntax:**

**show ospf spf log** <options>

| Option | Description |
|---|---|
| - | Without any option, the command displays the SPF log of all instances. |
| instance <instance-name> | Displays SPF log for the specified instance. |
| <afi> | Displays SPF log information for the specified address family. Supported AFI values are 'ipv4' and 'ipv6'. You can display SPF log information for the specified address family using filters such as interface name, detail, and instance. |

Example 1: OSPF SPF Result for the default instance

```
supervisor@rtbrick: op> show ospf spf log
Instance: default
  Router ID: 0.0.0.0
    Schedule timestamp: 2026-06-05 14:48:00, Area ID: 0.0.0.0, LSA type: Unknown
    Reason: ABR Status change, Back off timer: 50, LSA count: 1
    LS ID: 0.0.0.0, Number of schedule request: 2
    SPF start time: 2026-06-05 14:48:00, Number of nodes: 0, Number of links: 0
    Number of stub links: 0, SPF init time: 0 Timer Value, SPF run time: 0 Timer Value
    Router LSA change count: -, Network LSA change count: -
    Prefix changes: 1, Sequence number: 1
  Router ID: 0.0.0.0
```

```
    Schedule timestamp: 2026-06-05 14:48:00, Area ID: 0.0.0.1, LSA type: Unknown
    Reason: ABR Status change, Back off timer: 50, LSA count: 1
    LS ID: 0.0.0.0, Number of schedule request: 2
    SPF start time: 2026-06-05 14:48:00, Number of nodes: 0, Number of links: 0
    Number of stub links: 0, SPF init time: 0 Timer Value, SPF run time: 0 Timer Value
    Router LSA change count: -, Network LSA change count: -
    Prefix changes: 1, Sequence number: 2
  Router ID: 0.0.0.0
    Schedule timestamp: 2026-06-05 14:48:00, Area ID: 0.0.0.0, LSA type: Unknown
    Reason: ABR Status change, Back off timer: 50, LSA count: 2
    LS ID: 0.0.0.0, Number of schedule request: 3
    SPF start time: 2026-06-05 14:48:00, Number of nodes: 1, Number of links: 0
    Number of stub links: 0, SPF init time: 79us, SPF run time: 1048us
    Router LSA change count: 1, Network LSA change count: -
    Prefix changes: 1, Sequence number: 3
  Router ID: 0.0.0.0
    Schedule timestamp: 2026-06-05 14:48:00, Area ID: 0.0.0.1, LSA type: Unknown
    Reason: ABR Status change, Back off timer: 50, LSA count: 2
    LS ID: 0.0.0.0, Number of schedule request: 3
    SPF start time: 2026-06-05 14:48:00, Number of nodes: 1, Number of links: 0
    Number of stub links: 0, SPF init time: 13us, SPF run time: 939us
    Router LSA change count: 1, Network LSA change count: -
    Prefix changes: 1, Sequence number: 4
  Router ID: 198.51.100.11
    Schedule timestamp: 2026-06-05 14:48:00, Area ID: 0.0.0.0, LSA type: Router
    Reason: Router LSA change, Back off timer: 200, LSA count: 1
    LS ID: 198.51.100.11, Number of schedule request: 1
    SPF start time: 2026-06-05 14:48:00, Number of nodes: 1, Number of links: 0
    Number of stub links: 1, SPF init time: 15us, SPF run time: 892us
    Router LSA change count: 1, Network LSA change count: -
    Prefix changes: 1, Sequence number: 5
  Router ID: 198.51.100.11
    Schedule timestamp: 2026-06-05 14:48:00, Area ID: 0.0.0.1, LSA type: Router
    Reason: Router LSA change, Back off timer: 200, LSA count: 1
    LS ID: 198.51.100.11, Number of schedule request: 1
    SPF start time: 2026-06-05 14:48:00, Number of nodes: 1, Number of links: 0
    Number of stub links: 2, SPF init time: 10us, SPF run time: 788us
    Router LSA change count: 1, Network LSA change count: -
    Prefix changes: 1, Sequence number: 6
  Router ID: 198.51.100.11
    Schedule timestamp: 2026-06-05 14:48:00, Area ID: 0.0.0.1, LSA type: Link
    Reason: Unknown map - (value) 0xa, Back off timer: 200, LSA count: 1
    LS ID: 1.8.0.0, Number of schedule request: 1
    SPF start time: 2026-06-05 14:48:00, Number of nodes: 1, Number of links: 0
    Number of stub links: 0, SPF init time: 14us, SPF run time: 874us
    Router LSA change count: 1, Network LSA change count: -
    Prefix changes: 1, Sequence number: 7
  Router ID: 198.51.100.11
    Schedule timestamp: 2026-06-05 14:48:00, Area ID: 0.0.0.0, LSA type: Link
    Reason: Unknown map - (value) 0xa, Back off timer: 200, LSA count: 1
    LS ID: 1.8.144.1, Number of schedule request: 1
```

```
    SPF start time: 2026-06-05 14:48:00, Number of nodes: 1, Number of links: 0
    Number of stub links: 0, SPF init time: 13us, SPF run time: 209us
    Router LSA change count: 1, Network LSA change count: -
    Prefix changes: 1, Sequence number: 8
  Router ID: 198.51.100.21
    Schedule timestamp: 2026-06-05 14:48:04, Area ID: 0.0.0.1, LSA type: Opaque-Area
    Reason: RI LSA change, Back off timer: 5000, LSA count: 4
    LS ID: 4.0.0.0, Number of schedule request: 4
    SPF start time: 2026-06-05 14:48:09, Number of nodes: 2, Number of links: 2
    Number of stub links: 4, SPF init time: 56us, SPF run time: 790us
    Router LSA change count: 4, Network LSA change count: -
    Prefix changes: 2, Sequence number: 9
  Router ID: 198.51.100.11
    Schedule timestamp: 2026-06-05 14:48:40, Area ID: 0.0.0.1, LSA type: Link
    Reason: Unknown map - (value) 0xa, Back off timer: 50, LSA count: 1
    LS ID: 1.8.0.0, Number of schedule request: 1
    SPF start time: 2026-06-05 14:48:40, Number of nodes: 1, Number of links: 0
    Number of stub links: 0, SPF init time: 17us, SPF run time: 227us
    Router LSA change count: 1, Network LSA change count: -
    Prefix changes: 1, Sequence number: 10
  Router ID: 198.51.100.11
    Schedule timestamp: 2026-06-05 14:48:40, Area ID: 0.0.0.0, LSA type: Link
    Reason: Unknown map - (value) 0xa, Back off timer: 50, LSA count: 1
    LS ID: 1.8.144.1, Number of schedule request: 1
    SPF start time: 2026-06-05 14:48:40, Number of nodes: 1, Number of links: 0
    Number of stub links: 0, SPF init time: 26us, SPF run time: 202us
    Router LSA change count: 1, Network LSA change count: -
    Prefix changes: 1, Sequence number: 11
  Router ID: 198.51.100.21
    Schedule timestamp: 2026-06-05 14:48:43, Area ID: 0.0.0.0, LSA type: Link
    Reason: Unknown map - (value) 0xa, Back off timer: 5000, LSA count: 4
    LS ID: 1.28.144.1, Number of schedule request: 4
    SPF start time: 2026-06-05 14:48:48, Number of nodes: 2, Number of links: 0
    Number of stub links: 0, SPF init time: 25us, SPF run time: 220us
    Router LSA change count: 2, Network LSA change count: -
    Prefix changes: 2, Sequence number: 12
  Router ID: 198.51.100.21
    Schedule timestamp: 2026-06-05 14:48:48, Area ID: 0.0.0.0, LSA type: Router
    Reason: Router LSA change, Back off timer: 5000, LSA count: 3
    LS ID: 0.0.0.0, Number of schedule request: 3
    SPF start time: 2026-06-05 14:48:53, Number of nodes: 3, Number of links: 4
    Number of stub links: 0, SPF init time: 32us, SPF run time: 709us
    Router LSA change count: 4, Network LSA change count: 1
    Prefix changes: 2, Sequence number: 13
```

#### OSPF Route

Displays OSPF routing table information.

**Syntax:**

**show ospf route** <options>

| Option | Description |
|---|---|
| - | Without any option, the command displays the OSPF route information for all instances. |
| area <area-id> | OSPF route information for the given area. |
| instance <instance-name> | OSPF route information for the given instance. |
| instance <instance-name> <afi> | Displays OSPF route information for the specified address family and instance. Supported AFI values are 'ipv4' and 'ipv6'. |
| instance <instance-name> <afi> <safi> | Displays OSPF route information for the specified address family (AFI), sub-address family (SAFI), and instance. Supported AFI values are 'ipv4' and 'ipv6'. Supported SAFI values are 'unicast', and 'labeled-unicast''. |
| instance <instance-name> label <label> | Displays OSPF route information for the specified label and instance. |
| instance <instance-name> mpls unicast label <label> \| type <type> | Displays OSPF route information for the specified MPLS unicast label or type for the instance. |
| prefix <ip> | Displays OSPF route information for the specified match prefix. |
| type <type> | Displays information for OSPF route type. The route types include external-type-1, external-type-2, inter-area, intra-area, and ospf-direct. |
| <afi> <safi> | Displays OSPF route information for the specified address family (AFI) and sub-address family (SAFI). Supported AFI values are 'ipv4' and 'ipv6'. Supported SAFI values are 'unicast', and 'labeled-unicast'' |
| label <label> | Displays information about the OSPF-labeled routes. |
| mpls unicast <label \| type> | Displays information about OSPF MPLS routes. |
| area-border | Displays the OSPF Area Border Router (ABR) information. Refer to section "3.1.7. OSPF Route ABR" for the interface configuration details. |
| autonomous-system-boundary | Displays Autonomous System Border Router information. Refer to section "3.1.7. OSPF Route ABR" for the interface configuration details. |

Example: OSPF route information for the default instance

```
supervisor@rtbrick: op> show ospf route
Instance: default, AFI: ipv4, SAFI: unicast
  Prefix                    Area            Type            Cost      Next Hop        Interface
  192.0.2.0/30              0.0.0.1         ospf-direct     10000     n/a             local
  198.51.100.11/32          0.0.0.0         ospf-direct     10000     n/a             local
  198.51.100.12/32          0.0.0.1         ospf-direct     10000     n/a             local
```

```
  198.51.100.21/32                          external-type-1  16777215  192.0.2.2       ifl-0/0/2/0
  198.51.100.22/32          0.0.0.1         intra-area       20000     192.0.2.2       ifl-0/0/2/0
Instance: default, AFI: ipv4, SAFI: labeled-unicast
  Prefix                    Area            Type             Cost      Next Hop        Interface
  198.51.100.11/32          0.0.0.0         ospf-direct      10000     n/a             local
Instance: default, AFI: ipv6, SAFI: unicast, Instance ID: 0
  Prefix                                      Area             Type             Cost      Next Hop
Interface
  2001:db8:0:1::11/128                        0.0.0.0          ospf-direct      10000     n/a
local
  2001:db8:0:1::12/128                        0.0.0.1          ospf-direct      10000     n/a
local
  2001:db8:0:1::21/128                        0.0.0.0          intra-area       20000     fe80::9819:2c00:6499:472d
ifl-0/0/2/100
  2001:db8:0:1::22/128                        0.0.0.0          inter-area       20000     fe80::9819:2c00:6499:472d
ifl-0/0/2/100
  2001:db8:1:1::/64                           0.0.0.1          ospf-direct      10000     n/a
local
  2001:db8:2:1::/64                           0.0.0.0          ospf-direct      10000     n/a
local
Instance: default, AFI: ipv6, SAFI: labeled-unicast, Instance ID: 0
  Prefix                    Area            Type             Cost      Next Hop        Interface
  2001:db8:0:1::11/128      0.0.0.0         ospf-direct      10000     n/a             local
  2001:db8:0:1::21/128      0.0.0.0         intra-area       20000     n/a             ifl-0/0/2/100
Instance: default, AFI: mpls, SAFI: unicast
  Label                Type            Next Hop                                    Interface
  label:10100          ospf-direct     n/a                                         local
  label:10100,bos:1    ospf-direct     n/a                                         local
Instance: default, AFI: mpls, SAFI: unicast, Instance ID: 0
  Label                Type            Next Hop                                    Interface
  label:10101          ospf-direct     n/a                                         local
  label:10101,bos:1    ospf-direct     n/a                                         local
  label:10201          intra-area      fe80::9819:2c00:6499:472d                   ifl-0/0/2/100
  label:10201,bos:1    intra-area      fe80::9819:2c00:6499:472d                   ifl-0/0/2/100
```

#### OSPF Route Area Border

Displays the OSPF Area Border Router (ABR) information.

### Syntax:

```
show ospf route area-border
```

Example: OSPF Route ABR information

```
supervisor@rtbrick: op> show ospf route area-border
  198.51.100.11   0        198.51.100.11       -|-|E|B      local             -
  198.51.100.11   0        198.51.100.11       -|-|E|B      local             -
  198.51.100.21   10000    198.51.100.21       -|-|E|B      ifl-0/0/2/0       192.0.2.2
  198.51.100.11   0        198.51.100.11       -|-|-|B      local             -
  198.51.100.21   10000    198.51.100.21       -|-|-|B      ifl-0/0/2/100     fe80::9819:2c00:6499:472d
```

#### OSPF Route Autonomous System Boundary

Displays Autonomous System Boundary Router information.

### Syntax:

**show ospf route autonomous-system-boundary**

Example: OSPF Route ASBR information

```
supervisor@rtbrick: op> show ospf route autonomous-system-boundary
Instance: default, Address family: ipv4
  Node ID         Cost     Advertising Router  Flags        Interface         Nexthop
  198.51.100.21   10000    198.51.100.21       -|-|E|B      ifl-0/0/2/0       192.0.2.2
```

#### OSPF LSA Request List

Displays the list of all link-state advertisements (LSAs) requests that have been sent or received by a router.

**Syntax:**

**show ospf request-list** <options>

| Option | Description |
|---|---|
| - | Without any option, this command displays the list of all link-state advertisement (LSA) requests that have been sent from the router. |
| detail | Provides detailed information on the requests that have been sent from the router. |
| area <area-id> | OSPF request-list information for the given area. |
| instance <instance-name> | OSPF request-list information for the given instance. |
| <afi> | Displays request-list information for the specified address family. Supported AFI values are 'ipv4' and 'ipv6'. |

Example 1: OSPF LSA requests sent to a neighbor

```
supervisor@rtbrick: op> show ospf request-list
No pending LSA requests
```

Example 2: Detailed information for OSPF LSA requests sent to a neighbor

```
supervisor@rtbrick: op> show ospf request-list detail
No pending LSA requests
```

#### OSPF Transmission List

Displays the list of all LSAs waiting to be re-sent or transmitted from the router.

**Syntax:**

**show ospf transmit-list** <option>

| Option | Description |
|---|---|
| - | Without any option, this command displays the transmit list of all link-state advertisements (LSA). |
| area <area-id> | OSPF transmit-list information for the given area. |
| instance <instance-name> | OSPF transmit-list information for the given instance. |
| <afi> | Displays transmit-list information for the specified address family. Supported AFI values are 'ipv4' and 'ipv6'. |

Example: OSPF LSA requests waiting to be transmitted.

```
supervisor@rtbrick: op> show ospf transmit-list
No pending LSA transmissions
```

#### OSPF Statistics

Displays OSPF statistics information.

**Syntax:**

**show ospf statistics** <options>

| Option | Description |
|---|---|
| interface <interface-name> | Displays packet statistics information for the specified interface. |
| interface <interface-name> detail | Displays detailed packet statistics information for the specified interface. |
| neighbor <Neighbor-address> | Displays packet statistics information for the specified neighbor. |
| neighbor <Neighbor-address> <detail> | Displays detailed packet statistics information for the specified neighbor. |
| <afi> | Displays packet statistics information for the specified address family. Supported AFI values are 'ipv4' and 'ipv6'. |

```
supervisor@rtbrick: op> show ospf statistics interface ifl-0/0/2/0 detail
Instance: default, Address family: ipv4
  Interface: ifl-0/0/2/0, Peer address: 192.0.2.2
    Hello packet:
      Received packets: 9, Sent packets: 9, Total errors: 0, Unsupported option: 0
      Area mismatch: 0, Area type option mismatch: 0, Dead interval mismatch: 0
      Hello interval mismatch: 0, Mask mismatch: 0, Self router ID: 0
```

```
      Obj add fail: 0, Source address mismatch: , Misc: 0
    DD packet:
      Received packets: 2, Sent packets: 3, Total errors: 0, Unsupported option: 0
      Invalid state packet received: 0, MTU mismatch: 0, DD obj add fail: 0, Misc: 0, Negotiation fail: 0
      Master bit mismatch: 0, Exchange state init pkt: 0, Capabilities mismatch: 0
      Expected seq mismatch: 0, Full state init pkt: 0, Invalid lsa: 0, Invalid external lsa: 0,
      LSDB Absent: 0, Lsa lookup fail: 0, Ls req add fail: 0
    LS request packet:
      Received packets: 1, Sent packets: 1, Total errors: 0, Invalid LSA type: 0
      Invalid state packet received: 0, LSA lookup error: 0, LSA lookup fail: 0
      LSA obj add fail: 0, Misc: 0
    LS update packet:
      Received packets: 3, Sent packets: 2, Total errors: 0, Invalid LSA type: 0
      Zero length LSA: 0, LSA length exceeded: 0, LSA checksum fail: 0
      Invalid state packet received: 0, LSA obj add fail: 0, Invalid LSA payload: 0, Misc: 0
    Ls ack packet:
      Received packets: 2, Sent packets: 3, Total errors: 0, LSA obj add fail: 0
      Invalid state packet received: 0, Misc: 0
    Sanity errors:
      Payload max len error: 0, Payload min len error: 0, Invalid version: 0
      Invalid auth data len: 0, Auth data missing: 0, Invalid packet min len: 0
      Invalid area ID: 0, Invalid network mask: 0, Authentication fail: 0
```

### OSPF Clear Commands

#### Clear OSPF Neighbor

Clear OSPF neighbor state information.

**Syntax:**

**clear ospf neighbor** <options>

| Option | Description |
|---|---|
| - | Without any option, the command clears all the OSPF neighbors. |
| instance <instance-name> | Clears OSPF neighbor information for the specified instance. |
| instance <instance-name> <afi> | Clears OSPF neighbor for the specified address family. Supported AFI values are 'ipv4' and 'ipv6'. Supported AFI values are 'ipv4' and 'ipv6'. When using an IPv6 address family with OSPFv3, an instance ID ranging from 0 to 31 must be specified. |
| instance <instance-name> <afi> area <area-id> | Clears OSPF neighbor for the specified area of the specified instance and address family. |
| instance <instance-name> <afi> area <area-id> interface <interface-name> | Clears OSPF neighbor for the specified interface for the specified area and address family of the specified instance. |
| force | Forcefully clears all the OSPF neighbors. This may impact DR/BDR election. |
| force instance <instance-name> | Forcefully clears the neighbor for the specified instance. |

| Option | Description |
| --- | --- |
| `force instance <instance-name> <afi>` | Forcefully clears OSPF neighbor for the specified address family. |
| `force instance <instance-name> <afi> area <area-id>` | Forcefully clears OSPF neighbor for the specified area of the specified instance and address family. |
| `force instance <instance-name> <afi> area <area-id> interface <interface-name>` | Forcefully clears OSPF neighbor for a specific interface and area of the specified instance and address family. |

Example:

```
supervisor@rtbrick>SPINE01: cfg> clear ospf neighbor instance default ipv6 instance-id 0 area 0 interface ifl-0/0/0/1
Triggered clear neighbor successfully
```

API To access the Operational State API that corresponds to this CLI, click include::common:partial$operational_api_links.adoc[tag=clear_ospf_neighbor].

#### Clear OSPF Statistics

Clear the OSPF statistics for all instances or a specified instance.

**Syntax:**

**clear ospf statistics** <options>

| Option | Description |
| --- | --- |
| `-` | Without any option, the command clears all the OSPF statistics. |
| `instance <instance-name>` | Clears OSPF statistics information for the specified instance. |
| `instance <instance-name> <afi>` | Clears OSPF statistics for the specified address family of the specified instance. Supported AFI values are 'ipv4' and 'ipv6'. When using an IPv6 address family with OSPFv3, an instance ID ranging from 0 to 31 must be specified. |
| `instance <instance-name> <afi> area <area-id>` | Clears OSPF statistics for the specified area of the specified instance and address family. |
| `instance <instance-name> <afi> area <area-id> interface <interface-name>` | Clears OSPF statistics for the specified interface for the specified area of the specified instance and address family. |

Example:

```
supervisor@rtbrick>SPINE01: cfg> clear ospf statistics instance default ipv6 instance-id 0 area 0.0.0.0 interface ifl-
0/0/0/1
Cleared statistics for all ipv6 neighbors under Instance [default] Area [0.0.0.0] Interface [ifl-0/0/0/1]
```

#### Clear OSPF Database

Clear the OSPF database for all instances.

**Syntax:**

**clear ospf database** <options>

| Option | Description |
|---|---|
| - | Without any option, the command clears all the OSPF database information. |
| <afi> | Clears OSPF database for the specified address family. Supported AFI values are 'ipv4' and 'ipv6'. Supported AFI values are 'ipv4' and 'ipv6'. When using an IPv6 address family with OSPFv3, an instance ID ranging from 0 to 31 must be specified. |
| <afi> instance <instance-name> | Clears OSPF database for the specified area of the specified instance and address family. |

Example:

```
supervisor@rtbrick>SPINE01: cfg> clear ospf database ipv6 instance default instance-id 0
Triggered clear database successfully
```

# 2.8. LDP

## 2.8.1. LDP Overview

Label distribution protocol (LDP) is the most commonly used protocol in the MPLS network. It generates and distributes labels and thus helps in MPLS packet switching and forwarding. By using LDP, label-switching routers in an MPLS network can exchange label mapping information to create label-switched paths (LSPs) for switching data packets. RtBrick FullStack (RBFS) supports Dual-stack, which means LDP can exchange FEC-label bindings over either IPv4 or IPv6 networks.

#### Peer Discovery

LDP sends UDP multicast hello packets to discover its neighbors and establishes neighbor adjacency with other directly connected label switch routers (LSRs). The hello message is periodically sent on LDP-enabled interfaces.

#### Session Establishment

After peer discovery, “initialization messages” are sent to each other. In these messages, the session Parameters are sent to each other. The LDP sessions are maintained by periodic keep-alive message.

After the LDP neighbors are discovered, the TCP session is established and the LDP FSM is triggered,

and LDP session becomes operational. LSRs start exchanging label mapping information with each other.

#### Dual-stack LDP

By default, RBFS is dual-stack capable, which means it can exchange IPv4/IPv6 FEC bindings over IPv4/IPv6 media (LDP over IPv4/IPv6).

To enable or disable a particular address family in RBFS, use "status <enable|disable>" CLI. For details, see LDP Address Family Configuration.

When LDP is enabled on an interface that supports both IPv4 and IPv6, LDP will start exchanging IPv4 hellos. To send an IPv6 hello, the source IPv6 address must be configured. For details about configuring the source address, see LDP Instance Configuration.

By default, both IPv4 and IPv6 hello will use the same transport preference as IPv6, but this can be changed by using the "connection-preference <ipv4|ipv6>" CLI. For details, see LDP Instance Configuration.

The following points should be noted regarding this functionality:

**Source address**:

- The source address specifies the transport address, either IPv4 or IPv6, to be used by the LDP protocol.
- When the source address is not configured, only an IPv4 transport connection will be established. As a result, only IPv4 FECs will be installed. To install IPv6 FECs, the source IPv6 address needs to be configured.

**Connection Preference**:

- When LDP is configured for dual-stack operation, the transport connection preference is IPv6 by default, and LDP will establish TCP connections over IPv6. To establish an IPv4 LDP session, the connection preference must be set to IPv4, and LDP will distribute IPv4 and IPv6 FECs over IPv4 connections.

#### Targeted Label Distribution Protocol (T-LDP)

RBFS supports Targeted Label Distribution Protocol (T-LDP) that enables a router to establish an LDP session with a router that is not directly connected. T-LDP is commonly used in MPLS networks to facilitate the distribution of labels across various segments of the network.

#### Label Generation

LDP generates label bindings for the IP addresses of the LDP-enabled loopback interfaces and then advertises them to all neighbors.

#### Label Management Modes

#### Label Advertisement Mode

LDP supports the Downstream Unsolicited feature in RBFS, where label bindings are advertised to all upstream neighbors. By default, label advertisement operates in the Downstream Unsolicited mode.

#### Label Distribution Control Mode

LDP supports the Ordered Label Distribution Control, where an LSR will initiate the transmission of the label mapping only for the prefix for which it has a label mapping from the next hop of the prefix or for which it is an egress.

#### Label Retention Mode

LDP supports the Liberal Label Retention Mode where all the label mapping advertisements for all routes received from all the LDP neighbors are retained.

### Supported LDP Standards

| RFC Number | Description |
|---|---|
| RFC 5036 | LDP Specification<br><br>The following modes are supported by RBFS for the features listed in RFC 5036:<br><br>• Label advertisement: Downstream Unsolicited mode is supported but not Downstream on Demand mode.<br>• Label distribution control: Ordered mode is supported, but not Independent mode.<br>• Label retention: Liberal mode is supported, but not Conservative mode. |
| RFC 5283 | LDP Extension for Inter-Area Label Switched Paths (LSPs) |
| RFC 5443 | LDP IGP Synchronization |
| RFC 7552 | IPv6 Dual-Stack |

RFC and draft compliance are partial except as specified.

### Supported LDP Features

The following LDP features are supported in this release of RBFS:

- Support for the following label management modes.
  - Downstream unsolicited mode in label advertisement
  - Ordered mode in the label distribution control
  - Liberal mode in label retention
- Loop detection

- Targeted Label Distribution Protocol (T-LDP)
- Inter-area support
- Tracking IGP metric
- IGP LDP synchronization
- LDP Dual-stack support
- LDP TCP authentication
- LDP redistribution
- LDP policy configuration

**Supported Platforms**

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

## 2.8.2. LDP Configuration

**LDP Configuration Hierarchy**

The diagram below illustrates the LDP configuration hierarchy.

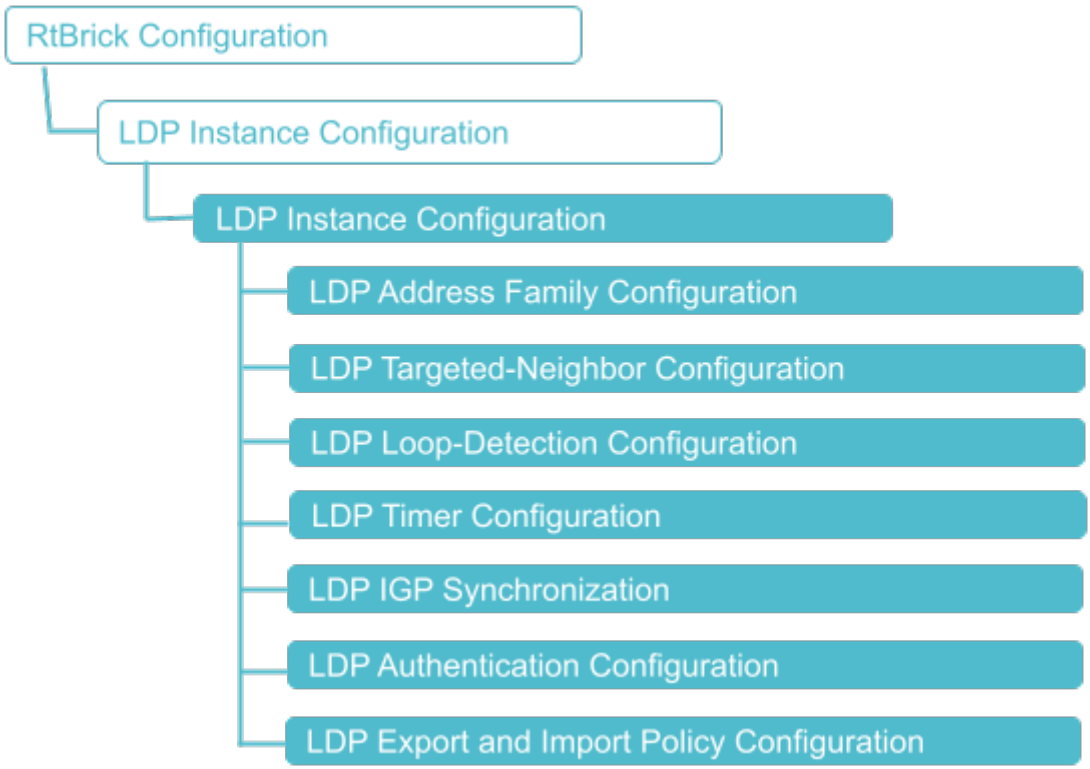



*Figure 18. LDP Configuration Hierarchy*

**Configuration Syntax and Commands**

The following sections describe the LDP configuration syntax and commands.

#### LDP Instance Configuration

At this configuration hierarchy, you configure LDP protocol parameters which are generic to the LDP instance.

**Syntax:**

**set instance** <instance-name> **protocol ldp** <attribute> <value>

| Attribute | Description |
|---|---|
| `<instance-name>` | Name of the LDP instance. |
| `interface <name>` | Name of the logical interface. |
| `router-id <router-id>` | Router identifier in IPv4 format. |
| `address-family <afi>` | Address family identifier (AFI). Supported values: ipv4, or ipv6. Refer to section LDP Address Family Configuration for LDP address family configuration details. |
| `connection-preference <afi>` | Specifies the connection preference for the TCP session. Supported values: ipv4, or ipv6. By default, IPv6 is used as the preferred TCP connection if an IPv6 source address is configured. Refer to section Dual-stack LDP for more information on the LDP Dual-stack behaviour. |
| `igp-synchronization <…>` | LDP IGP synchronization configuration. This option is supported only on interfaces running Intermediate System-to-System (IS-IS) or OSPFv2 processes. Refer to section LDP IGP Synchronization for LDP-IGP synchronization configuration details. |
| `source-address <ipv4\|ipv6> <source-address>` | Use the specified IP addresses (IPv4 or IPv6) as the transport address for the LDP session. For LDP over IPv6, the IPv6 source address is mandatory. Refer to section Dual-stack LDP for more information on the LDP Dual-stack behavior. |
| `loop-detection <…>` | The LDP loop detection feature enables LDP to detect loops during an LSP establishment. Refer to section LDP Loop Detection Configuration for the loop detection configuration details. |
| `protocol-preference (internal \| external) <preference-value>` | Specifies the protocol preference of routes learned by T-LDP ('external'), LDP ('internal'), or both. This preference is used to select routes learned from multiple protocols.<br><br>external: Range: 1 to 255. Default: 190. internal: Range: 1 to 255. Default: 9. |
| `timer <…>` | Specifies the Hello hold time, Hello interval, Keepalive hold time, and Keepalive interval. Refer to section LDP Timer Configuration for the LDP timer configuration details. |

| Attribute | Description |
|---|---|
| `session <ipv4\|ipv6> <address> `authentication-id <…>` | Specifies an IPv4 or IPv6 LDP session attributes to apply TCP authentication. Refer to section LDP Authentication Configuration. |
| `targeted-neighbor <ipv4\|ipv6> <neighbor-address> <source-address>` | Specifies the IPv4 or IPv6 address of the targeted LDP neighbor. Refer to section LDP Targeted Neighbor Configuration. |
| `session <ipv4\|ipv6> <address> export-policy\|import-policy <…>` | Specifies an IPv4 or IPv6 LDP session attributes to apply import/export policy configurations. Refer to section LDP Import and Export Policy Configuration. |

Example: LDP Instance Configuration

This following configuration example sets the LDP router ID and enables LDP on various interfaces for the `default` instance.

```
set instance default protocol ldp router-id 198.51.100.1
set instance default protocol ldp interface ifl-0/0/0/1
set instance default protocol ldp interface ifl-0/0/0/100
set instance default protocol ldp interface ifl-0/0/0/101
set instance default protocol ldp interface ifl-0/0/1/102
set instance default protocol ldp interface ifl-0/0/2/1
set instance default protocol ldp interface ifl-0/0/3/1
set instance default protocol ldp interface lo-0/0/0/1
set instance default protocol ldp interface lo-0/0/0/2
set instance default protocol ldp interface lo-0/0/0/3
set instance default protocol ldp interface lo-0/0/0/4
set instance default protocol ldp interface lo-0/0/0/5
commit
```

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ldp
{
    "rtbrick-config:ldp": {
      "router-id": "198.51.100.1",
      "interface": [
        {
          "name": "ifl-0/0/0/1"
        },
        {
          "name": "ifl-0/0/0/100"
        },
        {
          "name": "ifl-0/0/0/101"
        },
        {
          "name": "ifl-0/0/1/102"
        },
        {
```

```
          "name": "ifl-0/0/2/1"
        },
        {
          "name": "ifl-0/0/3/1"
        },
        {
          "name": "lo-0/0/0/1"
        },
        {
          "name": "lo-0/0/0/2"
        },
        {
          "name": "lo-0/0/0/3"
        },
        {
          "name": "lo-0/0/0/4"
        },
        {
          "name": "lo-0/0/0/5"
        }
      ]
    }
  }
```

The following configuration example sets the administrative preference values for external and internal LDP routes within the default instance.

```
set instance default protocol ldp protocol-preference external 150
set instance default protocol ldp protocol-preference internal 8
commit
```

```
supervisor@rtbrick>LEAF01: cfg> show config instance default protocol ldp protocol-
preference
{
  "rtbrick-config:protocol-preference": {
    "external": 150,
    "internal": 8
  }
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### LDP Targeted Neighbor Configuration

The targeted neighbor configuration establishes an LDP neighbor with a remote neighbor that is not directly connected.

**Syntax:**

**set instance** <instance> **protocol ldp targeted-neighbor** <attribute> <value>

| Attribute | Description |
|---|---|
| `ipv4 <neighbor-address>` | IPv4 address of the remote LDP neighbor. |
| `ipv4 <neighbor-address> <source-address>` | Neighbhor address to which ldp hello is sent |
| `ipv4 <neighbor-address> <source-address> hold-time <hold-time>` | Targeted hold time, <0-65535>, default 45s |
| `ipv4 <neighbor-address> <source-address> interval <interval>` | Targeted hello messages interval, <0-65535>, default 15s |
| `ipv4 <neighbor-address> <source-address> neighbor-type <receiver \|sender>` | Specifies the neighbor type, that is, sender or receiver. |
| `ipv6 <neighbor-address>` | IPv6 address of the remote LDP neighbor. |
| `ipv6 <neighbor-address> <source-address>` | Neighbhor address to which ldp hello is sent |
| `ipv6 <neighbor-address> <source-address> hold-time <hold-time>` | Targeted hold time, <0-65535>, default 45s |
| `ipv6 <neighbor-address> <source-address> interval <interval>` | Targeted hello messages interval, <0-65535>, default 15s |
| `ipv6 <neighbor-address> <source-address> neighbor-type <receiver \|sender>` | Specifies the neighbor type, that is, sender or receiver. |

The following example configurates the targeted LDP neighbors for both IPv4 and IPv6 addresses within the default instance.

```
set instance default protocol ldp targeted-neighbor ipv4 198.51.100.1 198.51.100.2
set instance default protocol ldp targeted-neighbor ipv4 198.51.100.5 198.51.100.2
set instance default protocol ldp targeted-neighbor ipv6 2001:DB8::1 2001:DB8::2
set instance default protocol ldp targeted-neighbor ipv6 2001:DB8::5 2001:DB8::2
```

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ldp targeted-
neighbor
{
  "rtbrick-config:targeted-neighbor": {
    "ipv4": [
      {
        "neighbor-address": "198.51.100.1",
        "source-address": "198.51.100.2"
      },
      {
```

```
        "neighbor-address": "198.51.100.5",
        "source-address": "198.51.100.2"
      }
    ],
    "ipv6": [
      {
        "neighbor-address": "2001:DB8::1",
        "source-address": "2001:DB8::2"
      },
      {
        "neighbor-address": "2001:DB8::5",
        "source-address": "2001:DB8::2"
      }
    ]
  }
}
```


To access the RESTCONF API that corresponds to this CLI, click here.

#### LDP Address Family Configuration

The address-family command allows you to enable the address families that LDP will route and configure settings that are specific to that address family.

**Syntax:**

**set instance** <instance-name> **protocol ldp address-family** <attribute> <value>

| Attribute | Description |
|---|---|
| `<afi>` | Address family identifier (AFI). Supported AFI values: ipv4, ipv6. |
| `<safi>` | Subsequent address family identifier (SAFI). Supported SAFI value: labeled-unicast.<br>Configuring SAFI is now mandatory during the redistribution configuration or when enabling/disabling status. |
| `<afi> <safi> status <enable\|disable>` | Enable or disable address family. By default, both IPv4 and IPv6 address families are enabled, as LDP supports dual stack. Refer to section Dual-stack LDP for more information on the LDP Dual-stack behavior. |
| `<afi> <safi> redistribute <source>` | Specifies the source from which the routes are to be redistributed. The available options include `direct`, `ipoe`, `isis`, `ospf`, `ppp`, and `static`. |

| Attribute | Description |
| --- | --- |
| `<afi> <safi> redistribute <source> policy <policy>` | Specifies the name of the policy map. The redistribute attach point allows routes from other sources to be advertised by LDP. The policy can be applied only to the routes that are redistributed from other sources to LDP. |

Example 1: LDP Address Family Configuration

```
set instance default protocol ldp address-family ipv6 labeled-unicast status enable
commit
```

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ldp address-
family
{
  "rtbrick-config:address-family": [
    {
      "afi": "ipv6",
      "safi": "labeled-unicast",
      "status": "enable"
    }
  ]
}
```

Example 2: LDP Redistribution Configuration

```
set instance default protocol ldp address-family ipv4 redistribute direct
commit
```

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ldp address-
family ipv4 redistribute direct
{
  "rtbrick-config:redistribute": [
    {
      "source": "direct"
    }
  ]
}
```

Example 3: LDP Policy Configuration

```
set instance default protocol ldp address-family ipv4 redistribute direct policy
filter-link-address
commit
```

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ldp address-
family ipv4 redistribute
{
  "rtbrick-config:redistribute": [
    {
      "source": "direct",
      "policy": "filter-link-address"
    }
  ]
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### LDP Loop Detection Configuration

The LDP loop detection feature enables LDP to detect loops during an LSP establishment.

**Syntax:**

**set instance** <instance-name> **protocol ldp loop-detection** <attribute> <value>

| Attribute | Description |
|---|---|
| `hop-count <hop-count>` | Specifies the hop count limit for loop detection. Range: 0-255. Default: 32. |
| `status <enable\|disable>` | Enables or disables loop detection. By default, this option is disabled. When this option is enabled, both hop count and path vector are enabled. |
| `vector-length <vector-length>` | Specifies the path vector length limit for loop detection. Range: 0-255. Default: 32. |

Example 1: LDP Loop Detection Configuration

```
set instance default protocol ldp loop-detection enable true
set instance default protocol ldp loop-detection hop-count 64
set instance default protocol ldp loop-detection vector-length 64
commit
```

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ldp loop-
detection
{
    "rtbrick-config:loop-detection": {
      "enable": "true",
      "hop-count": 64,
      "vector-length": 64
    }
```

```
  }
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### LDP Timer Configuration

Specify the hello timer and hold-down timer for LDP adjacency. Similarly, specify the keepalive and keepalive timeout settings for the LDP session.

**Syntax:**

**set instance** <instance-name> **protocol ldp timer** <attribute> <value>

| Attribute | Description |
|---|---|
| `hello hold-time <hold-time>` | Specifies the hello hold-time interval in seconds before declaring a neighbor to be down. Range: 0-65535. Default: 15. |
| `hello interval <interval>` | Specifies the hello messages interval in seconds. Range: 0-65535. Default: 5. |
| `session keepalive-interval <keepalive-interval>` | Specifies the session keepalive messages interval in seconds. Range: 1-65535. Default: 10. |
| `session keepalive-timeout <keepalive-timeout>` | Specifies the session keepalive timeout in seconds before declaring a session to be down. Range: 1-65535. Default: 30. |

Example 1: LDP Timer Configuration

```
set instance default protocol ldp timer hello interval 10
set instance default protocol ldp timer hello hold-time 20
set instance default protocol ldp timer session keepalive-interval 3000
set instance default protocol ldp timer session keepalive-timeout 5000
commit
```

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ldp timer
{
    "rtbrick-config:timer": {
      "hello": {
        "interval": 10,
        "hold-time": 20
      },
      "session": {
        "keepalive-interval": 3000,
        "keepalive-timeout": 5000
      }
    }
  }
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### LDP IGP Synchronization

Synchronization between LDP and the underlying interior gateway protocol (IGP) ensures that the LDP path is fully established before the IGP path is used for forwarding traffic. LDP IGP synchronization is supported only on interfaces running Intermediate System-to-System (IS-IS) or OSPFv2 processes.

**Syntax:**

**set instance** <instance-name> **protocol ldp igp-synchronization** <attribute> <value>

| Attribute | Description |
|---|---|
| `hold-timer <hold-timer>` | Specifies the hold-timer in seconds to limit how long the IGP session must wait before declaring the LDP synchronization. Range: 0-60. Default: 10. |

Example 1: LDP IGP Synchronization Configuration

```
set instance default protocol ldp igp-synchronization hold-timer 60
commit
```

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ldp igp-
synchronization
{
    "rtbrick-config:igp-synchronization": {
     hold-timer": 60,
    }
  }
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### LDP Authentication Configuration

To meet the security requirements of LDP sessions, configure LDP authentication.

**Syntax:**

**set instance** <instance-name> **protocol ldp session ipv4|ipv6** <address> **authentication-id** <authentication-id>

| Attribute | Description |
|---|---|
| `<address>` | Specifies the transport IP address of the peer. |

| Attribute | Description |
|---|---|
| `<authentication-id>` | Authentication Tuple Identifier |

Example 1: LDP Authentication Configuration

```
set instance default protocol ldp session ipv4 address 198.51.100.2 authentication-id
auth_id_1
commit
```

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ldp session
ipv4
{
  "rtbrick-config:ipv4": [
    {
      "address": "198.51.100.2",
      "authentication-id": "auth_id_1"
    }
  ]
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### TCP Authentication Configuration

In the instance TCP authentication hierarchy, you can optionally enable MD5 or HMAC-SHA-1-96, HMAC-SHA-256-128, or AES-128-CMAC-96 authentication. Authentication is not configured for LDP directly, but for the TCP sessions used by LDP.

**Syntax:**

**set instance** <instance> **tcp authentication** <authentication-id> <attribute> <value>

| Attribute | Description |
|---|---|
| `<authentication-id>` | Authentication identifier |
| `type <type>` | Authentication identifier such as MD5 or HMAC-SHA-1-96,HMAC-SHA-256-128 or AES-128-CMAC-96 |
| `key1-id <key1-id>` | Key ID1 of the receiver |
| `key1-encrypted-text <key1-encrypted-text>` | Encrypted text of key1 |
| `key1-plain-text <key1-plain-text>` | Plain text of key1 |
| `key2-id <key2-id>` | Key ID2 of the receiver |

| Attribute | Description |
|---|---|
| key2-encrypted-text <key2-encrypted-text> | Encrypted text of key2 |
| key2-plain-text <key2-plain-text> | Plain text of key2 |

Example: LDP TCP Authentication Configuration

```
set instance default tcp authentication auth_id_1 type MD5
set instance default tcp authentication auth_id_1 key1-id 1
set instance default tcp authentication auth_id_1 key1-encrypted-text
"$2a6fd7db50a18a9f1f16b5c5b4214fab0"
commit
```

```
supervisor@rtbrick>SPINE01: cfg> show config instance default tcp authentication
auth_id_1
{
  "rtbrick-config:authentication": [
    {
      "authentication-id": "auth_id_1",
      "type": "MD5",
      "key1-id": 1,
      "key1-encrypted-text": "$2a6fd7db50a18a9f1f16b5c5b4214fab0"
    }
  ]
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### LDP Import and Export Policy Configuration

**Syntax:**

**set instance** <instance-name> **protocol ldp session ipv4|ipv6** <address> <attribute> <value>

| Attribute | Description |
|---|---|
| <address> | Specifies the IPv4 or IPv6 address. |
| export-policy <export-policy> | Export policy identifier |
| import-policy <import-policy> | Import policy identifier |

Example 1: LDP Export Configuration

```
set instance default protocol ldp session ipv4 address 198.51.100.2 export-policy exp-
policy1
```

```
commit
```

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ldp session
ipv4
{
  "rtbrick-config:ipv4": [
    {
      "address": "198.51.100.2",
      "export-policy": "exp-policy1"
    }
  ]
}
```

Example 3: LDP Import Configuration

```
set instance default protocol ldp session ipv4 address 198.51.100.2 import-policy imp-
policy1
commit
```

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ldp session
ipv4
{
  "rtbrick-config:ipv4": [
    {
      "address": "198.51.100.2",
      "import-policy": "imp-policy1"
    }
  ]
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

### 2.8.3. LDP Operational Commands

#### LDP Show Commands

The LDP show commands provide detailed information about the LDP protocol operations.

##### LDP Summary

**Syntax:**

**show ldp summary** <options>

| Option | Description |
|---|---|
| - | Without any option, the command displays the LDP summary information for all instances. |
| instance <instance-name> | Displays LDP summary information about the specified instance. |

Example: LDP summary for the default instance

```
supervisor@rtbrick: op> show ldp summary
Instance: default
  General information:
    LDP identifier: 198.51.100.11:0, Version: 1
    FEC resolution: Best match
    Protocol preference: 9
    LSR ID: 198.51.100.11
    IPv4 Status: True
    IPv6 Status: True
    L2VPN VPLS-VPWS Status: False
  Modes:
    Advertisement mode: Downstream Unsolicited
    Advertisement control mode: Ordered
    Label retention mode: Liberal
  Capabilities:
    IPv6 address family: -  , Graceful restart: False
    Loop detection: False
      Hop count: -, Vector length: -
  Timers:
    Adjacency:
      Hello: 5s, Holdtime: 15s
    Targeted adjacency:
      Hello: 15s, Holdtime: 45s
    Session:
      Keepalive: 10s, Holdtime: 30s
  Statistics:
    Adjacency:
      Link adjacency: 4, Targeted adjacency: 0
    Session:
      Session in non-existent: 0, Session in initialized: 0
      Session in opensent: 0, Session in openconfirm: 0
      Session in operational: 1
```

API To access the Operational State API that corresponds to this CLI, click here.

#### LDP Neighbor

**Syntax:**

**show ldp neighbor** <options>

| Option | Description |
| --- | --- |
| - | Without any option, this command displays information about LDP neighbors. |
| detail | Detailed information about the LDP neighbors. |
| instance <instance-name> | Displays LDP neighbor information about the specified instance. |
| instance <instance-name> detail | Displays detailed LDP neighbor information about the specified instance. |
| instance <instance-name> ldp-id <ldp-id> | Displays LDP neighbor information about the specified LDP identifier and instance. |
| interface <name> | Displays LDP neighbor information about the specified interface. |
| interface <name> detail | Displays detailed LDP neighbor information about the specified interface. |
| ldp-id <ldp-id> | Displays LDP neighbor information about the specified LDP identifier. |

Example 1: Summary view of all the LDP neighbors

```
supervisor@rtbrick: op> show ldp neighbor
Instance: default
  Interface/Peer      LDP ID              Transport IP              Up Since                  Expires
  ifl-0/0/2/0         198.51.100.21:0     198.51.100.21             Fri Jun 05 14:49:56       in 13s
  ifl-0/0/2/100       198.51.100.21:0     198.51.100.21             Fri Jun 05 14:49:56       in 13s
  ifl-0/0/2/0         198.51.100.21:0     2001:db8:0:1::21          Fri Jun 05 14:49:56       in 13s
  ifl-0/0/2/100       198.51.100.21:0     2001:db8:0:1::21          Fri Jun 05 14:49:56       in 13s
```

Example 2: Detailed view of all the LDP neighbors

```
supervisor@rtbrick: op> show ldp neighbor detail
Instance: default
  LDP-Identifier: 198.51.100.21:0, Interface: ifl-0/0/2/0, Type: Link
  Negotiated holdtime: 15000, Expiry time: 12s 975097us
  Local link address: 192.0.2.1, Peer link address: 192.0.2.2
  Local transport address: 198.51.100.11, Peer transport address: 198.51.100.21
  Local holdtime: 15, Peer holdtime: 15, Up since: Fri Jun 05 14:49:56
  Local transport preference : ipv4, Peer transport preference : ipv4
  Last transition time: Fri Jun 05 14:50:06 GMT +0000 2026
Instance: default
  LDP-Identifier: 198.51.100.21:0, Interface: ifl-0/0/2/100, Type: Link
  Negotiated holdtime: 15000, Expiry time: 12s 975170us
  Local link address: 192.0.2.5, Peer link address: 192.0.2.6
  Local transport address: 198.51.100.11, Peer transport address: 198.51.100.21
  Local holdtime: 15, Peer holdtime: 15, Up since: Fri Jun 05 14:49:56
  Local transport preference : ipv4, Peer transport preference : ipv4
  Last transition time: Fri Jun 05 14:50:06 GMT +0000 2026
Instance: default
  LDP-Identifier: 198.51.100.21:0, Interface: ifl-0/0/2/0, Type: Link
  Negotiated holdtime: 15000, Expiry time: 12s 975086us
```

```
  Local link address: fe80::9819:2cff:fe06:b503, Peer link address: fe80::9819:2cff:fe99:472d
  Local transport address: 2001:db8:0:1::11, Peer transport address: 2001:db8:0:1::21
  Local holdtime: 15, Peer holdtime: 15, Up since: Fri Jun 05 14:49:56
  Local transport preference : ipv4, Peer transport preference : ipv4
  Last transition time: Fri Jun 05 14:50:06 GMT +0000 2026
Instance: default
  LDP-Identifier: 198.51.100.21:0, Interface: ifl-0/0/2/100, Type: Link
  Negotiated holdtime: 15000, Expiry time: 12s 975137us
  Local link address: fe80::9819:2c00:6406:b503, Peer link address: fe80::9819:2c00:6499:472d
  Local transport address: 2001:db8:0:1::11, Peer transport address: 2001:db8:0:1::21
  Local holdtime: 15, Peer holdtime: 15, Up since: Fri Jun 05 14:49:56
  Local transport preference : ipv4, Peer transport preference : ipv4
  Last transition time: Fri Jun 05 14:50:06 GMT +0000 2026
```

Example 3: Detailed view of a targeted LDP neighbor

```
supervisor@rtbrick: op> show ldp neighbor ldp-id 198.51.100.21:0
Instance: default
  LDP-Identifier: 198.51.100.21:0, Interface: ifl-0/0/2/0, Type: Link
  Negotiated holdtime: 15000, Expiry time: 11s 964088us
  Local link address: 192.0.2.1, Peer link address: 192.0.2.2
  Local transport address: 198.51.100.11, Peer transport address: 198.51.100.21
  Local holdtime: 15, Peer holdtime: 15, Up since: Fri Jun 05 14:49:56
  Local transport preference : ipv4, Peer transport preference : ipv4
  Last transition time: Fri Jun 05 14:50:06 GMT +0000 2026
Instance: default
  LDP-Identifier: 198.51.100.21:0, Interface: ifl-0/0/2/100, Type: Link
  Negotiated holdtime: 15000, Expiry time: 11s 964187us
  Local link address: 192.0.2.5, Peer link address: 192.0.2.6
  Local transport address: 198.51.100.11, Peer transport address: 198.51.100.21
  Local holdtime: 15, Peer holdtime: 15, Up since: Fri Jun 05 14:49:56
  Local transport preference : ipv4, Peer transport preference : ipv4
  Last transition time: Fri Jun 05 14:50:06 GMT +0000 2026
Instance: default
  LDP-Identifier: 198.51.100.21:0, Interface: ifl-0/0/2/0, Type: Link
  Negotiated holdtime: 15000, Expiry time: 11s 964117us
  Local link address: fe80::9819:2cff:fe06:b503, Peer link address: fe80::9819:2cff:fe99:472d
  Local transport address: 2001:db8:0:1::11, Peer transport address: 2001:db8:0:1::21
  Local holdtime: 15, Peer holdtime: 15, Up since: Fri Jun 05 14:49:56
  Local transport preference : ipv4, Peer transport preference : ipv4
  Last transition time: Fri Jun 05 14:50:06 GMT +0000 2026
Instance: default
  LDP-Identifier: 198.51.100.21:0, Interface: ifl-0/0/2/100, Type: Link
  Negotiated holdtime: 15000, Expiry time: 11s 964190us
  Local link address: fe80::9819:2c00:6406:b503, Peer link address: fe80::9819:2c00:6499:472d
  Local transport address: 2001:db8:0:1::11, Peer transport address: 2001:db8:0:1::21
  Local holdtime: 15, Peer holdtime: 15, Up since: Fri Jun 05 14:49:56
  Local transport preference : ipv4, Peer transport preference : ipv4
  Last transition time: Fri Jun 05 14:50:06 GMT +0000 2026
```

#### LDP Interface

**Syntax:**

**show ldp interface** <option>

| Option | Description |
|---|---|
| - | Without any option, this command displays information on all configured LDP interfaces. |
| <interface-name> | Specify the name of the interface. |
| detail | Detailed interface information. |

Example 1: Summary view of the specified LDP interface

```
supervisor@rtbrick: op> show ldp interface ifl-0/0/2/0
Instance: default
  Interface: ifl-0/0/2/0
    Primary IPv4 Address: 192.0.2.1, Primary IPv6 Address: fe80::9819:2cff:fe06:b503
    Session Hold: 30, Session Keepalive: 10
    Discovery Hello: 5, Discovery Hold: 15
    Neighbor count: 2, Transport preference: ipv4
    IPv4 enable: True, IPv6 enable: True, Cisco interop: False
```

Example 2: Summary view of all the LDP interfaces

```
supervisor@rtbrick: op> show ldp interface
Instance: default
  Interface                 Primary IPv4 Address      Primary IPv6 Address
  ifl-0/0/2/0               192.0.2.1                 fe80::9819:2cff:fe06:b503
  ifl-0/0/2/100             192.0.2.5                 fe80::9819:2c00:6406:b503
  lo-0/0/0/0                198.51.100.11             -
  lo-0/0/0/1                198.51.100.12             -
```

#### LDP Session

**Syntax:**

**show ldp session** <options>

| Option | Description |
|---|---|
| - | Without any option, this command displays a summary of LDP session information. |
| detail | Displays detailed information about the LDP sessions. |

| Option | Description |
|---|---|
| `instance <instance-name>` | Displays LDP session information about the specified instance. |
| `instance <instance-name> detail` | Displays detailed LDP session information about the specified instance. |
| `instance <instance-name> ldp-id <ldp-id>` | Displays LDP session information about the specified LDP identifier and instance. |
| `ldp-id <ldp-id>` | Displays LDP session information about the specified LDP identifier. |

Example 1: Summary view of the LDP session

```
supervisor@rtbrick: op> show ldp session
Instance: default
  LDP ID            Peer IP                   State             Up/Down           FECRcvd   FECSent
  198.51.100.21:0   198.51.100.21             Operational       0d:00h:00m:13s          8         8
```

Example 2: Detailed view of the LDP session

```
supervisor@rtbrick: op> show ldp session detail
Instance: default
  LDP Identifier: 198.51.100.21:0, Peer IP: 198.51.100.21, Local IP: 198.51.100.11
    Type: Link, State: Operational, Uptime: 0d:00h:00m:14s
    Last transition: Fri Jun 05 14:50:00 GMT +0000 2026, Flap count: 0
    Adjacency count: 2, Role: Passive
  Advertisement Mode:
    Peer: Downstream unsolicited, Local: Downstream unsolicited
    Negotiated: Downstream unsolicited
  Timers:
    Connect retry: 10s
    Peer keepalive interval: 10s, Local keepalive interval: 10s
    Peer keepalive timeout: 30s, Local keepalive timeout: 30s
    Negotiated keepalive interval: 10s
    Negotiated keepalive timeout: 30s
  Received messages:
    Initialization: 1, KeepAlive: 2, Notification: 0
    Address: 2, Address Withdraw: 0, Label Mapping: 8
    Label Withdraw: 0, Label Release: 0
  Sent messages:
    Initialization: 1, KeepAlive: 2, Notification: 0
    Address: 2, Address Withdraw: 0, Label Mapping: 8
    Label Withdraw: 0, Label Release: 0
  Capability:
    Typed WildCard FEC:
      Local Support: True, Peer Support: True, Negotiated: True
  Total received messages:
    Initialization: 1, KeepAlive: 2, Notification: 0
    Address: 2, Address Withdraw: 0, Label Mapping: 8
```

```
  Total sent messages:
    Initialization: 1, KeepAlive: 2, Notification: 0
    Address: 2, Address Withdraw: 0, Label Mapping: 8
    Label Withdraw: 0, Label Release: 0
```

In the case of a T-LDP session, the "Type" will be "Targeted".

To access the Operational State API that corresponds to this CLI, click here.

#### LDP Address

**Syntax:**

**show ldp address** <options>

| Option | Description |
|---|---|
| - | Without any option, this command displays a summary of all the interface addresses received from the LDP sessions. |
| instance <instance-name> | Displays LDP address information about the specified instance. |
| instance <instance-name> <afi> | Displays LDP address of the specified address family (AFI). Supported values: ipv4, ipv6. |
| instance <instance-name> ldp-id <ldp-id> | Displays LDP address information about the specified LDP identifier and instance. |
| ldp-id <ldp-id> | Displays LDP address information about the specified LDP identifier. |

Example: Summary view of all the LDP addresses

```
supervisor@rtbrick: op> show ldp address
Instance: default, LDP Identifier: 198.51.100.21:0, AFI: ipv4
  192.0.2.2
  192.0.2.6
  198.51.100.21
  198.51.100.22
Instance: default, LDP Identifier: 198.51.100.21:0, AFI: ipv6
  2001:db8:0:1::21
  2001:db8:0:1::22
  2001:db8:1:1::2
  2001:db8:2:1::2
  fe80::9819:2c00:6499:472d
  fe80::9819:2cff:fe99:472d
```

**LDP Binding**

**Syntax:**

**show ldp binding** <options>

| Option | Description |
|---|---|
| - | Without any option, this command displays a summary of all the LDP label bindings. |
| detail | Displays detailed LDP label binding information. |
| instance <instance-name> | Displays LDP label binding information of the specified instance. |
| instance <instance-name> <afi> | Displays LDP label binding information of the specified address family. Supported AFI value: IPv4, IPv6. |
| instance <instance-name> <afi> detail | Displays detailed LDP label binding information of the specified address family. |
| instance <instance-name> <afi> prefix <ip> | Displays LDP label binding information of the specified prefix and instance for the specified address family. |
| instance <instance-name> <afi> prefix <ip> detail | Displays detailed LDP label binding information of the specified prefix and instance for the specified address family. |
| received | Displays the LDP received label binding information of the LDP sessions. |
| received instance <instance-name> <afi> | Displays LDP received label binding information of the specified address family and instance. |
| received instance <instance-name> <afi> detail | Displays detailed LDP received label binding information of the specified address family and instance. |
| received instance <instance-name> <afi> ldp-id <ldp-id> | Displays LDP received label binding information of the specified instance. |
| received instance <instance-name> <afi> ldp-id <ldp-id> detail | Displays detailed LDP received label binding information of the specified LDP identifier and instance. |
| received <afi> | Displays LDP received label binding information of the specified address family. Supported AFI values: IPv4, IPv6. |
| received <afi> detail | Displays detailed LDP received label binding information of the specified address family. |
| received <afi> instance <instance-name> | Displays LDP received label binding information of the specified address family and instance. |
| received <afi> instance <instance-name> detail | Displays detailed LDP received label binding information of the specified address family and instance. |
| received <afi> instance <instance-name> ldp-id <ldp-id> | Displays LDP received label binding information of the specified address family, instance, and LDP identifier. |

| Option | Description |
|---|---|
| `received <afi> ldp-id <ldp-id>` | Displays LDP received label binding information of the specified address family and LDP identifier. |
| `received <afi> ldp-id <ldp-id> detail` | Displays detailed LDP received label binding information of the specified address family and LDP identifier. |
| `sent` | Displays the LDP sent label binding information of the LDP sessions. |
| `sent instance <instance-name> <afi>` | Displays LDP sent label binding information of the specified address family and instance. Supported AFI values: IPv4, IPv6. |
| `sent instance <instance-name> <afi> detail` | Displays detailed LDP sent label binding information of the specified address family and instance. |
| `sent instance <instance-name> <afi> prefix <ip>` | Displays LDP sent label binding information of the specified address family, instance, and prefix. |
| `sent instance <instance-name> <afi> prefix <ip> detail` | Displays detailed LDP sent label binding information of the specified address family, instance, and prefix. |
| `sent <afi>` | Displays LDP sent label binding information of the specified address family. Supported AFI values: IPv4, IPv6. |
| `sent <afi> detail` | Displays detailed LDP sent label binding information of the specified address family. |
| `sent <afi> instance <instance-name>` | Displays LDP sent label binding information of the specified address family and instance. |
| `sent <afi> instance <instance-name> detail` | Displays detailed LDP sent label binding information of the specified address family and instance. |
| `sent <afi> instance <instance-name> prefix <ip>` | Displays LDP sent label binding information of the specified address family, instance, and prefix. |
| `sent <afi> instance <instance-name> prefix <ip> detail` | Displays detailed LDP sent label binding information of the specified address family, instance, and prefix. |
| `sent <afi> prefix <ip>` | Displays LDP sent label binding information of the specified address family and prefix. |
| `sent <afi> prefix <ip> detail` | Displays detailed LDP sent label binding information of the specified address family and prefix. |

Example 1: Summary view of the LDP binding

```
supervisor@rtbrick: op> show ldp binding
Instance: default, AFI: ipv4, SAFI: labeled-unicast
  Prefix                                                In Label          Out Label        LDP ID             Status
  198.51.100.11/32                                             -                  3        -                  Best
                                                          524032                  -        198.51.100.21:0    Non-best
  198.51.100.12/32                                             -                  3        -                  Best
                                                          524032                  -        198.51.100.21:0    Non-best
  198.51.100.21/32                                             3             524032        198.51.100.21:0    Best
  198.51.100.22/32                                             3             524032        198.51.100.21:0    Best
Instance: default, AFI: ipv6, SAFI: labeled-unicast
```

```
  Prefix                                          In Label        Out Label       LDP ID            Status
  2001:db8:0:1::11/128                                   -                3       -                 Best
                                                    524033                -       198.51.100.21:0   Non-best
  2001:db8:0:1::12/128                                   -                3       -                 Best
                                                    524033                -       198.51.100.21:0   Non-best
  2001:db8:0:1::21/128                                   3           524033       198.51.100.21:0   Best
  2001:db8:0:1::22/128                                   3           524033       198.51.100.21:0   Best
```

Example 2: Summary view of the LDP binding for the specified prefix

```
supervisor@rtbrick: op> show ldp binding prefix 198.51.100.21/32
Instance: default, AFI: ipv4, SAFI: labeled-unicast
  Prefix                                          In Label        Out Label       LDP ID            Status
  198.51.100.21/32                                       3           524032       198.51.100.21:0   Best
```

#### LDP Route

### Syntax:

**show ldp route** <options>

| Option | Description |
|---|---|
| - | Without any option, this command displays a summary of LDP route information. |
| instance <instance-name> | Displays LDP route information for the specified instance. |
| instance <instance-name> <afi> | Displays LDP route information for the specified address family and instance. Supported AFI values: ipv4, ipv6, and mpls. |
| instance <instance-name> ipv4 prefix <ip> | Displays LDP route information for the specified address family of IPv4 prefix and instance. |
| instance <instance-name> ipv6 prefix <ip> | Displays LDP route information for the specified address family of IPv6 prefix and instance. |
| instance <instance-name> prefix <ip> | Displays LDP route information for the specified prefix and instance. |
| instance <instance-name> label <label> | Displays LDP route information for the specified MPLS label and instance. |
| instance <instance-name> mpls | Displays LDP route information about MPLS labels. |
| instance <instance-name> mpls label <label> | Displays LDP route information for the specified MPLS label and instance. |
| label <label> | Displays LDP route information for the specified MPLS label. |
| ipv4 | Displays LDP route information about the IPv4 address family. |
| ipv4 prefix <ip> | Displays LDP route IPv4 address family information for the specified prefix. |
| ipv6 | Displays LDP route information about the IPv6 address family. |

| Option | Description |
|---|---|
| `ipv6 prefix <ip>` | Displays LDP route IPv6 address family information for the specified prefix. |
| `mpls` | Displays LDP route information about MPLS labels. |
| `mpls label <label>` | Displays LDP route information for the specified MPLS label. |
| `prefix <ip>` | Displays LDP route information for the specified prefix address. |

Example: Summary view of the LDP route

```
supervisor@rtbrick: op> show ldp route
Instance: default, AFI: ipv4, SAFI: labeled-unicast
  Prefix               Out Label               Next Hop                                  Interface           Metric
  198.51.100.11/32     -                       -                                         -                   0
  198.51.100.12/32     -                       -                                         -                   0
  198.51.100.21/32     -                       192.0.2.2                                 -                   2000000
  198.51.100.22/32     -                       192.0.2.2                                 -                   2000000
Instance: default, AFI: ipv6, SAFI: labeled-unicast
  Prefix                                       Out Label               Next Hop
Interface           Metric
  2001:db8:0:1::11/128                         -                       -                                         -
0
  2001:db8:0:1::12/128                         -                       -                                         -
0
  2001:db8:0:1::21/128                         -                       fe80::9819:2cff:fe99:472d                 ifl-
0/0/2/0         2000000
  2001:db8:0:1::22/128                         -                       fe80::9819:2cff:fe99:472d                 ifl-
0/0/2/0         2000000
Instance: default, AFI: mpls, SAFI: unicast
  Label                Out Label               Next Hop                                  Interface           Metric
  524032               -                       192.0.2.2                                 -                   -
  524032,bos:1         -                       192.0.2.2                                 -                   -
  524033               -                       fe80::9819:2cff:fe99:472d                 ifl-0/0/2/0         -
  524033,bos:1         -                       fe80::9819:2cff:fe99:472d                 ifl-0/0/2/0         -
```

#### LDP Statistics

### Syntax:

**show ldp statistics** <options>

| Option | Description |
|---|---|
| `-` | Without any option, the command displays the LDP statistics for all instances. |
| `instance <instance-name>` | Displays LDP statistics information about the specified instance. |
| `instance <instance-name> ldp-id <ldp-id>` | Displays LDP statistics information about the specified LDP identifier and instance. |

Example: Summary view of the LDP statistics

```
supervisor@rtbrick: op> show ldp statistics
Instance: default, LDP ID: 198.51.100.21:0
  Received messages:
    Initialization: 1, KeepAlive: 2, Notification: 0
    Address: 2, Address Withdraw: 0, Label Mapping: 8
    Label Withdraw: 0, Label Release: 0
  Sent messages:
    Initialization: 1, KeepAlive: 2, Notification: 0
    Address: 2, Address Withdraw: 0, Label Mapping: 8
    Label Withdraw: 0, Label Release: 0
```

#### LDP TCP connection

**Syntax:**

**show ldp tcp connection** <options>

| Option | Description |
|---|---|
| - | Without any option, the command displays the TCP connections used by LDP for all instances. |
| detail | Detailed list view of the TCP connections. |
| detail instance <instance-name> | Detailed list view of the TCP connections of the specified instance. |
| instance <instance-name> | TCP connections summary of the specified instance. |

Example: Summary view of the LDP TCP connections

```
supervisor@rtbrick: op> show ldp tcp connection
Instance       Local IP            Remote IP           Local port     Remote port   State
default        198.51.100.11       198.51.100.21              646           58311   Established
```

### LDP Clear Commands

Clear commands allow resetting operational states.

#### Clear LDP Session

**Syntax:**

**clear ldp session** <options>

| Option | Description |
|---|---|
| all | Clears all the LDP sessions. |

| Option | Description |
|---|---|
| `all soft-in` | Sends route refresh to all neighbors to receive FEC bindings. |
| `all soft-out` | Re-advertises all the routes previously sent to the peers. |
| `instance <instance-name> all` | Clears all the LDP sessions for the specified instance. |
| `instance <instance> all soft-in` | Sends route refresh to all neighbors to receive FEC bindings for the specified instance. |
| `instance <instance> all soft-out` | Re-advertises all the routes previously sent to the peers for the specified instance. |
| `instance <instance-name> peer ldp-id <ldp-id>` | Clears the LDP session for the specified instance and peer LDP identifier. |
| `instance <instance> peer ldp-id <ldp-id> soft-in` | Sends route refresh to the specific peer to receive FEC bindings for the specified instance and peer ldp-id. |
| `instance <instance> peer ldp-id <ldp-id> soft-out` | Re-advertises all the routes previously sent to the specific peer for the specified instance and peer ldp-id. |

Example: The example below shows how to clear all the LDP sessions.

```
supervisor@rtbrick>SPINE01: op> clear ldp session all
LDP session cleared with instance default
supervisor@rtbrick>SPINE01: op>
```

API To access the Operational State API that corresponds to this CLI, click here.

#### Clear LDP Statistics

**Syntax:**

**clear ldp statistics** <options>

| Option | Description |
|---|---|
| `all` | Clears all the LDP statistics. |
| `instance` <instance-name> `all` | Clears all the LDP statistics for the specified instance. |
| `instance` <instance-name> `peer ldp-id` <ldp-id> | Clears the LDP statistics for the specified instance and peer LDP identifier. |

Example: The example below shows how to clear all the LDP statistics.

```
supervisor@rtbrick>SPINE01: op> clear ldp statistics all
LDP statistics cleared for instance default
supervisor@rtbrick>SPINE01: op>
```

**Clear LDP Neighbor**

**Syntax:**

```
clear ldp neighbor <options>
```

| Option | Description |
|---|---|
| `all` | Clears all the LDP neighbors. |
| `instance <instance-name>` | Clears the LDP neighbor for the specified instance. |

Example: The example below shows how to clear all the LDP neighbor.

```
supervisor@rtbrick>SPINE01: op> clear ldp neighbor all
LDP neighbor cleared with instance default
supervisor@rtbrick>SPINE01: op>
```

# 2.9. Policy

## 2.9.1. Policy Overview

Policies are rules that allow to control and modify the behavior of routing protocols such as BGP, IS-IS, OSPF, and other supported protocols. The policy framework is generic; it serves multiple purposes and applications. Policies are first created using a common policy configuration and then applied by attaching them to an application like a protocol.

The following diagram shows policy configuration options.

*Figure 19. Policy Configuration Options*

**Policy Components**

In RtBrick Full Stack, the policy implementation consists of 4 sub-components:

- Policy Repository
- Command Processing Module
- Policy Server, the policy generation and relationship management component
- Policy Client, the policy enforcement component

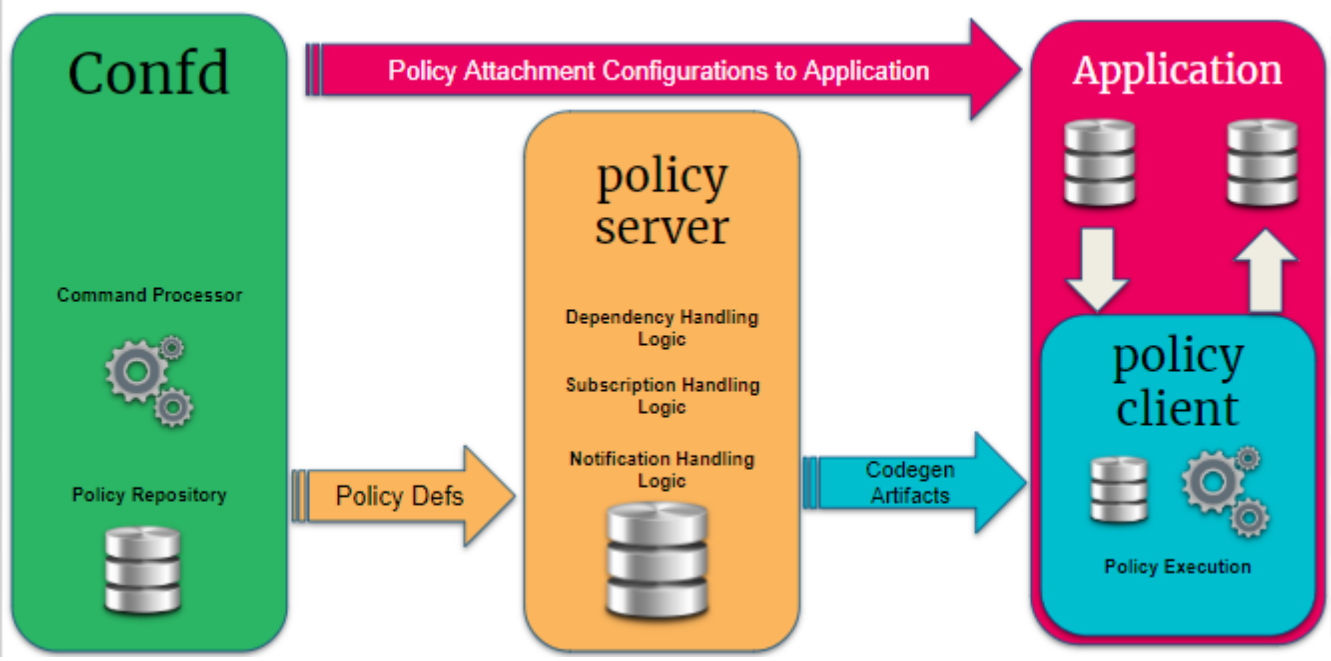



*Figure 20. Policy Components*

**Policy Repository**

The policy repository contains all tables related to policy and the associated list of match criteria.

**Command Processing Module**

The command processing module is part of the configuration daemon (confd), and it handles user interaction with the policy module. This is the back-end of the Command Line Interface (CLI) and JSON configuration that supports the policy configurations.

This module maps the user-defined configuration into the back-end policy object, which is used by the execution engine (after verification), and it ensures that the policy can be correctly executed. This module relays the user intent via related BDS tables to the policy server.

**Policy Server**

The policy server is a server component that manages all policy rules in the various policy tables and also code generation of the policies.

The following are the functionalities of the policy server:

- Parses the objects in the policy tables and is an execution engine that generates the code to build the policy rules for evaluation, the relationship between various objects, and relays the intent to the evaluation engine.
- Maintains relationships between various policy constructs such as policy statements, rules, ordinals, and lists.
- Tracks the attachment points so that when policies are modified, the appropriate clients are notified of the relevant new policies.
- Flattens the various relationships and generates a notification table that the clients subscribe to obtain notifications based on specific interest groups.

- Uses dependency table relationships to generate jobs to trigger code generation for various policy components.
- On code generation, the policy server updates a notification table that maintains the mapping between the policy server and the client interest groups. The notification table is a single point for the dissemination of information so that it can generate notifications for clients depending on their subscriptions for policy of interest.
- Policy server notification is generated for the policy clients. A notification is received from the notification table with metadata information that notifies the client if this is a new version of the policy or the original version of the policy. The client uses this information to enforce the policy evaluation and to decide on the version of the policy rule that should be used.

**Policy Client**

The policy client is a shared library component to which a client daemon, like BGP, IS-IS, OSPF, etc., links. This is the component that performs policy enforcement. It performs the following tasks:

- Links with client daemons like BGP, IS-IS, and OSPF.
- Contains a listener that gets notifications on the availability of a new policy rule that is generated by the policy server.
- Evaluates the compiled rule, and if there are any listeners/ interests, then notifies the components within the client daemon.
- Evaluates any policy configurations on the client daemon and invokes policy processing in response.

**Tables and Subscriptions**

The table below shows the various tables and their sharing across various policy components.

*Tables and Subscriptions*

| | | |
|---|---|---|
| Confd | global.policy.list.config<br>global.policy.list.entry.config<br>global.policy.match.rules.config<br>global.policy.statement.config<br>global.policy.ordinal.config<br>global.policy.mapping.list<br>global.policy.mapping.rules | Policy Statement is composed of one or more policy terms. Each term has a match action criteria. In the match and action criteria, either a single element or a list of elements are compared, and actions are taken. The actions include accept, deny, flow-control, etc. |
| policy.server | global.policy.dependency<br>global.<bds_name>.policy.subscription<br>global.<bds_name>.policy.notification | Policy Server subscribes to all the tables from `confd` and creates tables that track policy-entry and dependency and notifies clients after code generation. |

| | | |
|---|---|---|
| policy.client | global.<bds_name>.policy.shared.object.cache<br>global.<bds_name>.policy.subscription<br>global.<bds_name>.policy.context | Subscribes to code generation notifications application context and maintains cache of subscribed .so |

**Policy Building Blocks**

The figure below shows the building blocks of a policy.

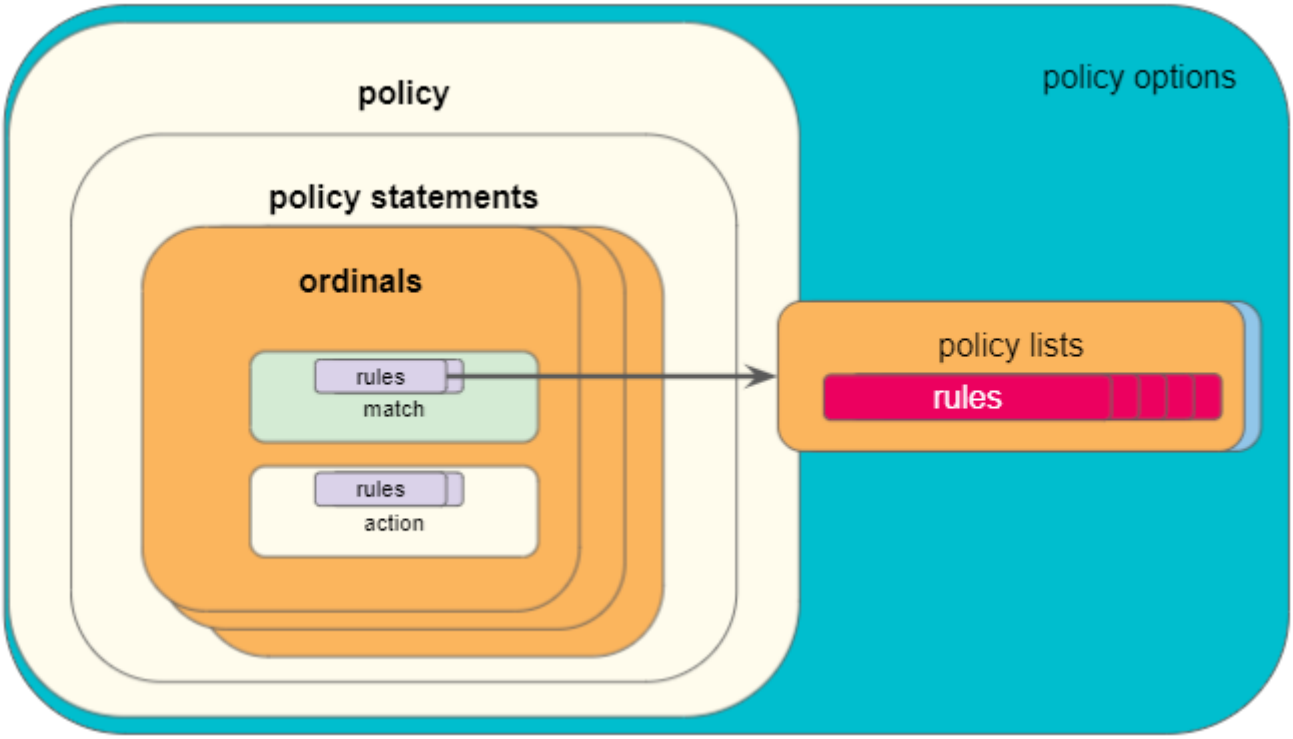



*Figure 21. Policy Building Blocks*

**Statements**

A policy is defined by a policy statement. A policy statement is a compound block of policy definition that consists of one or more set of rules called ordinals. A policy statement is exercised in the order defined. The statement name is a globally unique string that is used to identify the policy, and used by the applications.

The following diagram shows policy statement configuration parameters.

*Figure 22. Policy Statement Configuration*

**Ordinals**

A policy ordinal is the smallest block to represent a user policy intent and consists of rules for match and action blocks. The match blocks can either define single independent elements like AS path, IP prefix, IP addresses, community, and so on, or a list of the elements maintained in a different table.

- An ordinal must be a unique number within the scope of a statement which determines the order of the term execution within a policy statement.
- Only a maximum of 16000 ordinals is supported in a policy list.
- If no ordinal exist or configured, and if the policy is used, then all routes or objects will be denied.
- A match logic is defined per ordinal. In case of multiple match rules, it defines if all rules (`and`), or any of the rules (`or`) have to match.

**Match Rules**

Match rules define criteria to evaluate and select routes or other objects to which the policy is applied. One or more match rules compose a match block.

The following diagram shows policy match rule parameters.

*Figure 23. Policy Match Rules Configuration*

- If the match block results in a successful match ("true"), the corresponding action block will executed.
- If the match block result is unsuccessful ("false"), the action block will not be executed, and the next ordinal will be processed.

- If there is no match block, the action block will be executed.

In case of multiple match rules, the behaviour depends on the configured match logic:

- If the match logic is `or`, the match block result will be successful ("true") if any one rule matches. Otherwise, by default it is "false".
- If the match logic is `and`, the match block result will be successful ("true") if all rules match. Otherwise it is "false".

A match rule can refer to a single `discrete` value, or a list. A policy list is configured separately and referenced from the policy statement. In case of a list match, the behaviour is as follows:

- If any of the list entries matches the configured value, the match block result will be successful ("true"). Otherwise it is "false".
- If the list is defined but empty, the match rule result will be unsuccessful ("false").

#### Action Rules

Action rules define operations like `return-permit` or `return-deny`. The action block will be executed if there is a successful match. Otherwise, the next ordinal is processed. The action block can contain one or multiple action rules. In case of multiple action rules, all actions will be applied. The rules are executed in the order of the rule numbers.

If the action block is not defined, the policy result is `return-deny`.

The following diagram shows action rules configuration parameters.

*Figure 24. Policy Action Rules Configuration*

#### Lists

A policy list is a list of values that can be referenced by a match rule in a policy statement. If you have a number of values like for example route prefixes, it is more efficient to refer to a policy list instead of creating one match rule per prefix.

Policy lists are configured separately and can thereby be maintained more easily. Besides, policy lists can be referenced by multiple policy statements.

#### Conditions

Policy conditions are configured separately and can be used as an additional option in policy statements. A policy rule (ordinal) will only be executed if the condition is true. Conditional policies allow to make policy execution depended on certain states of the system, for example:

- Protocol neighbor states
- Presence or absence of a specific route and/or path attribute
- Number of routes in a routing table

One condition is supported per ordinal. A single condition can be attached to multiple policies.

#### Attachment Points

Policies define a set of rules that can be used for various purposes by various applications. Once policies have been created, they need to be applied to take effect. Attachment points describe the specific applications and processes to which policies can be applied.

The following diagram shows the BGP attachment points.

*Figure 25. BGP Attachment Points*

RBFS currently supports the following policy attachment points:

- Instance import/export - Policies attached to an instance at the address family level allow to control which routes will be imported into the instance and exported from the instance by BGP. Such import and export policies are commonly used with BGP L3VPNs.

The following diagram shows the BGP Global Instance attachment points.

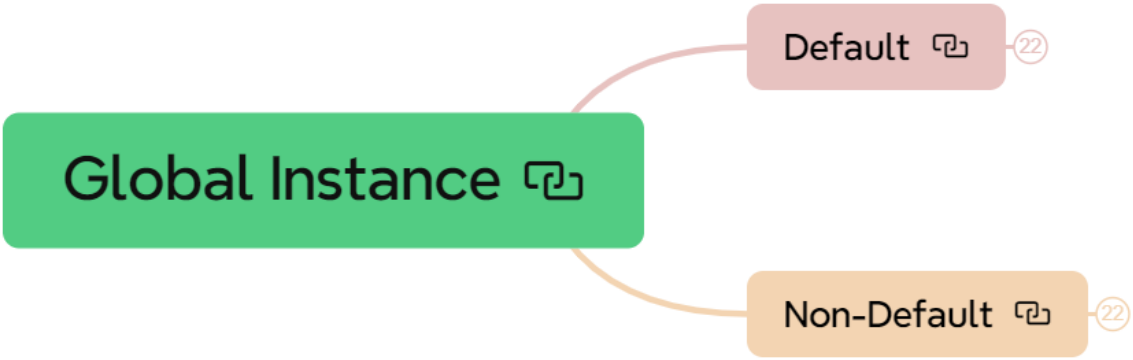



*Figure 26. BGP Global Instance Attachment Points*

The following diagram shows the BGP Global Instance default attachment points.

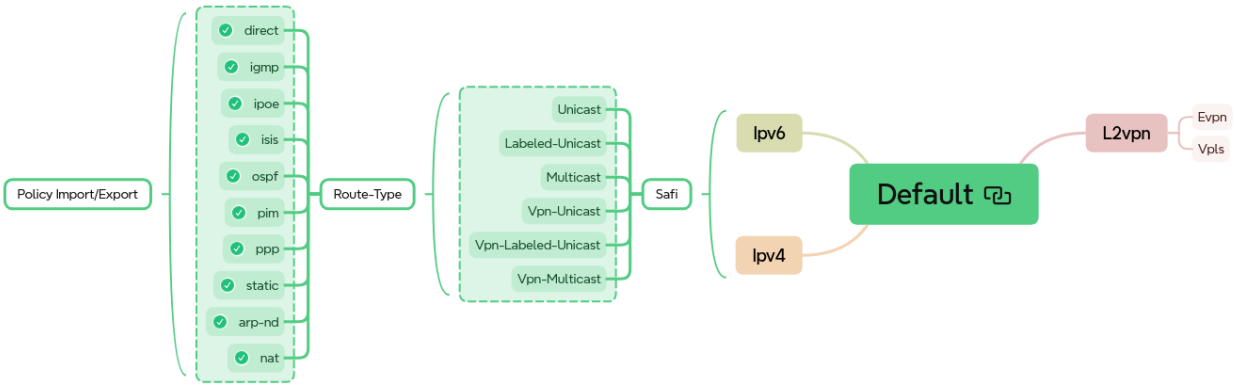



*Figure 27. BGP Global Instance Default Attachment Points*

The following diagram shows the BGP Global Instance non-default attachment points.

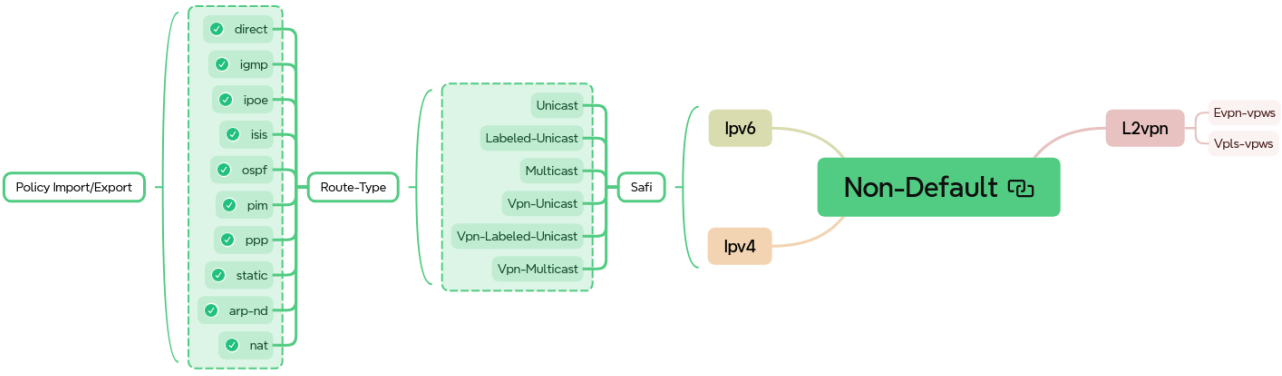



*Figure 28. BGP Global Instance Non-Default Attachment Points*

- BGP peer group import/export - Policies attached to BGP peer groups allow to define which routes will be advertised to and accepted from BGP peers. You can attach policies to both instances or peer groups to define the import and export behavior. You can also combine both attachment points, for example if some policy rules apply generically to the instance and some other rules specifically to a peer or peer group.

The following diagram shows BGP peer group attachment points.

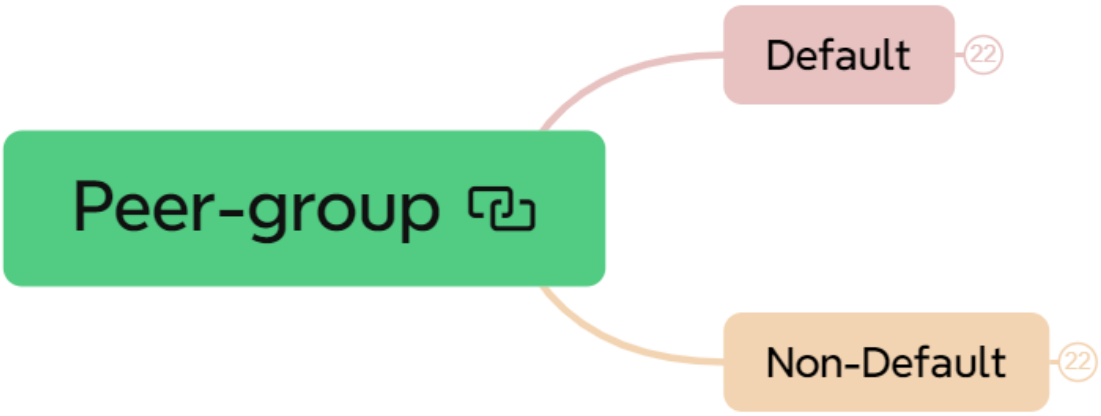



*Figure 29. BGP Peer Group Attachment Points*

The following diagram shows BGP peer group default attachment points.

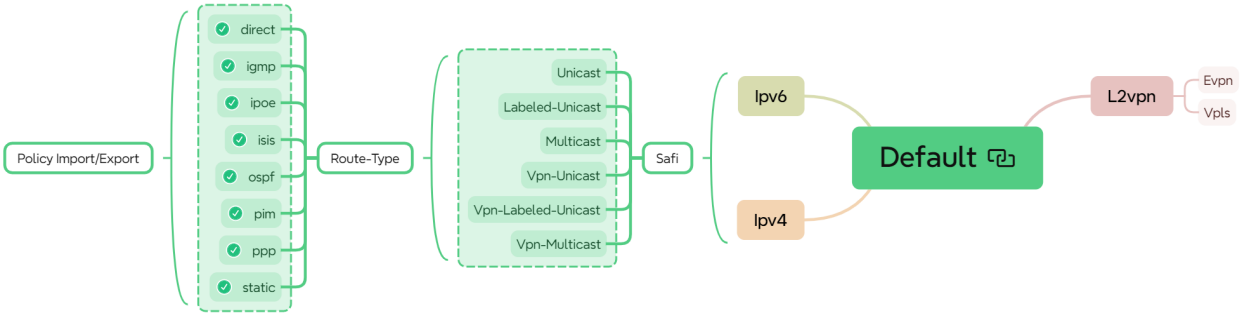



*Figure 30. BGP Peer Group Default Attachment Points*

The following diagram shows BGP peer group non-default attachment points.

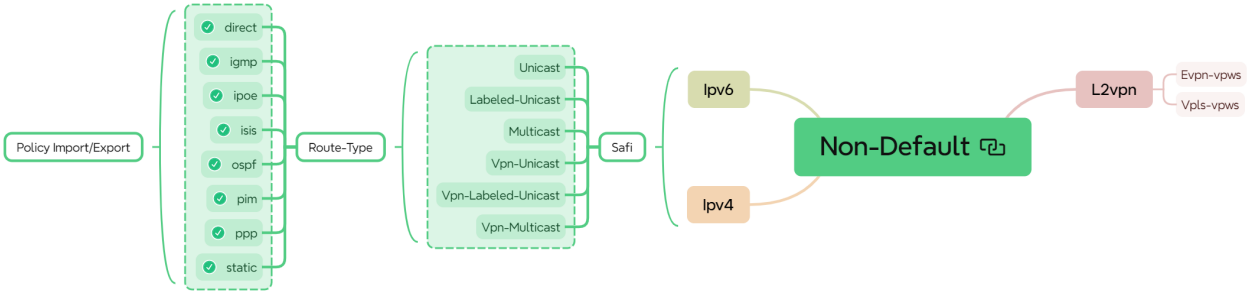



*Figure 31. BGP Peer Group Non-Default Attachment Points*

- BGP redistribution - You can attach policies to BGP redistribution to define which routes will be redistributed into the BGP process. This is useful if you would like to redistribute only a sub-set of a type of routes.

The following diagram shows BGP redistribution attachment points.

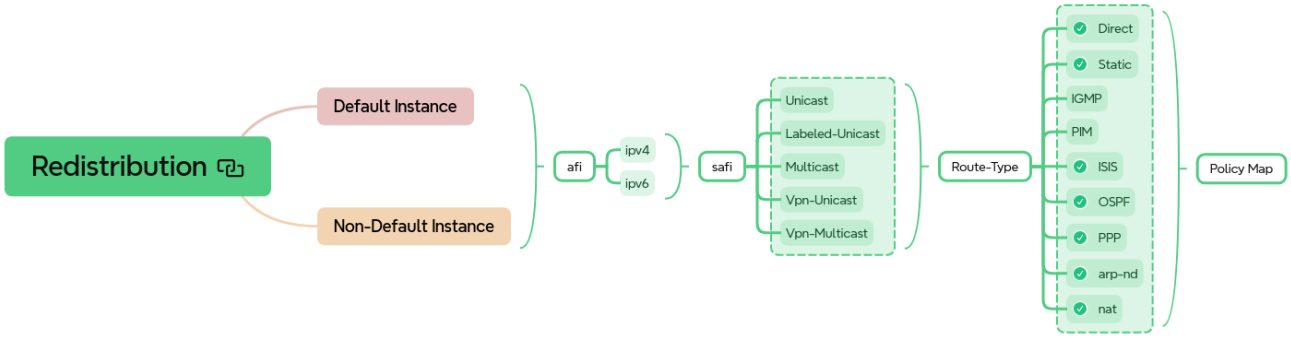



*Figure 32. BGP Redistribution Attachment Points*

- IS-IS redistribution - Policies can be attached to IS-IS redistribution to control which routes will be redistributed into the IS-IS process.

The following diagram shows IS-IS Redistribution attachment point.

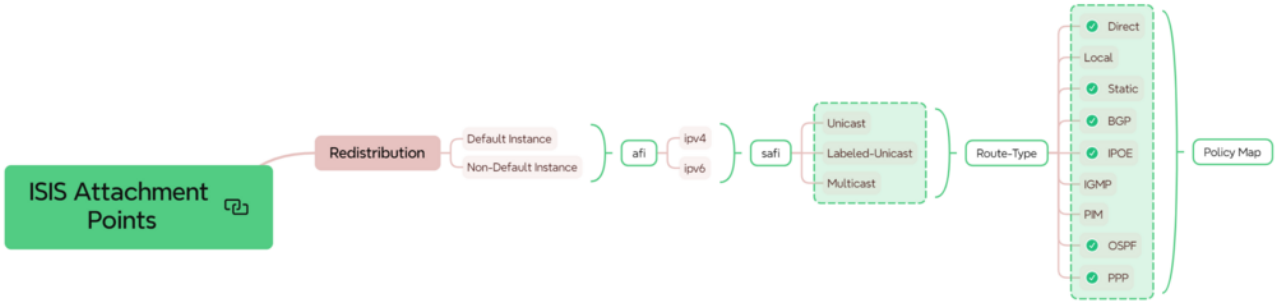



*Figure 33. IS-IS Redistribution Attachment Points*

- OSPFv2 redistribution - Policies can be attached to OSPFv2 redistribution to control which

routes will be redistributed into the OSPFv2 process.

The following diagram shows OSPF Redistribution attachment points.

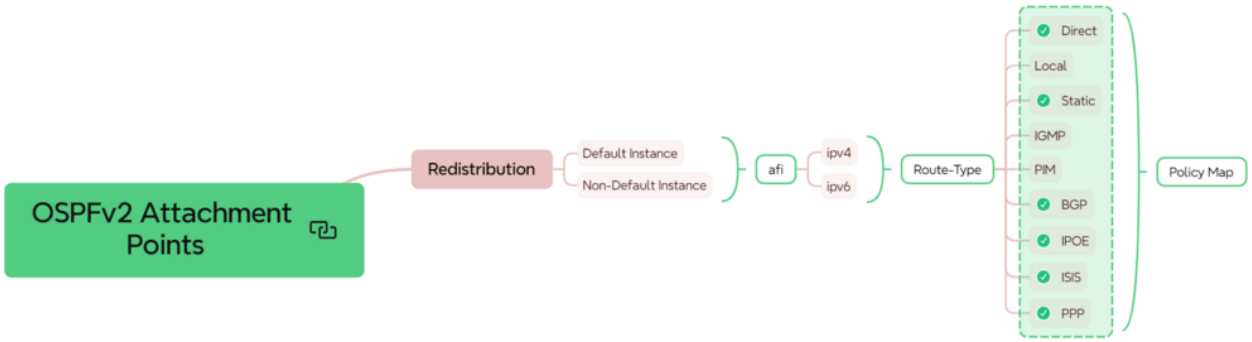



*Figure 34. OSPF Redistribution Attachment Points*

- LDP redistribution - Policies can be attached to the LDP redistribution to control which routes will be redistributed into the LDP process.

The following diagram shows LDP Redistribution attachment points.

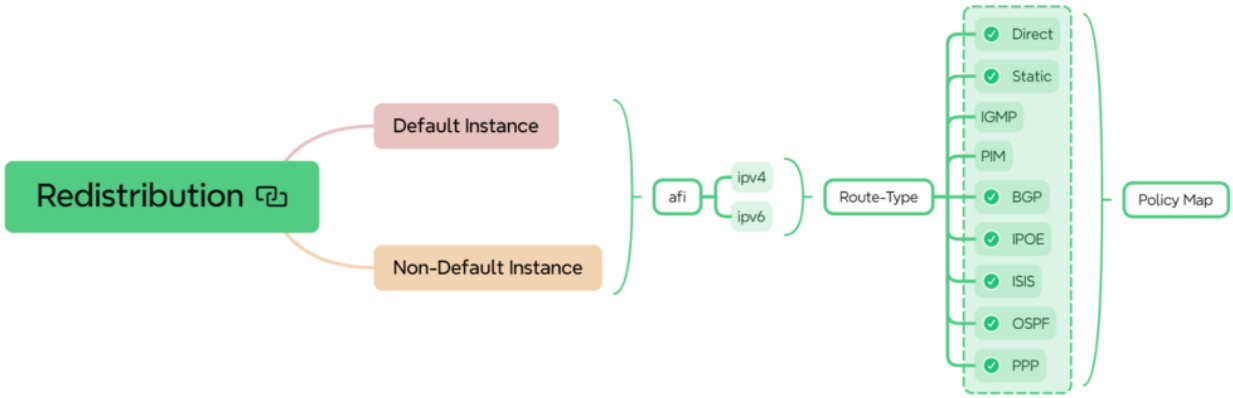



*Figure 35. LDP Redistribution Attachment Points*

The following diagram shows LDP peer group attachment points.

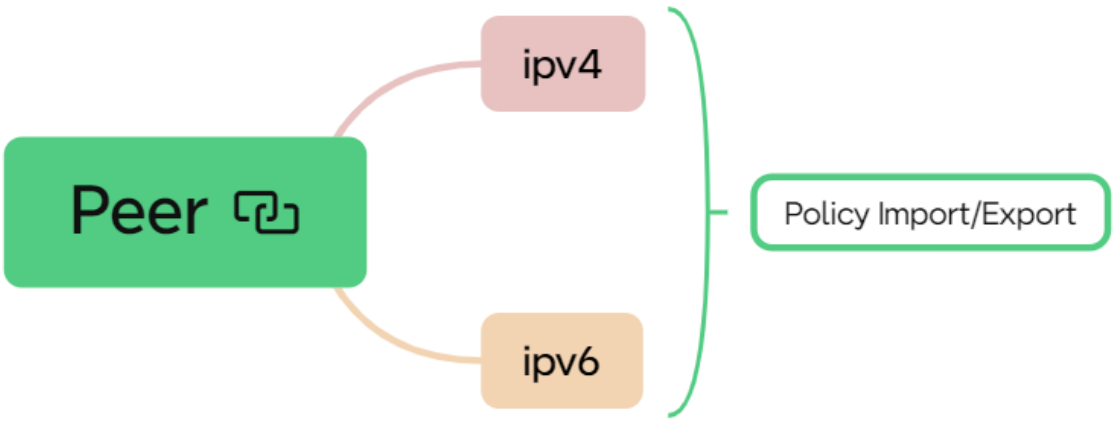



*Figure 36. LDP Peer Attachment Points*

- IGMP group filtering - You can apply policies to IGMP interface profiles. Such policies act as IGMP group filters when receiving IGMP Membership Report messages.
- IGMP SSM-mapping - Policies are further used to define SSM mapping. SSM mapping policies attached to IGMP interface profiles define how to translate IGMPv2 (*,G) to IGMPv3 (S,G) Reports.

The following diagram shows IGMP attachment points.

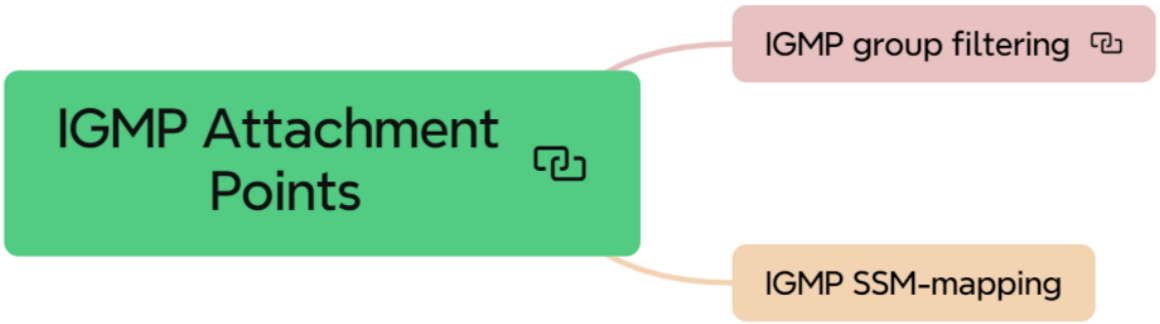



*Figure 37. IGMP Attachment Points*

**Policy Behaviour**

The default behaviour of a policy is `deny`. This means, if a route or any other object is subject to a policy, by default it will be marked as deny if the explicit action permit is not configured for the configured policy. For example import policy for a route, if the route or object successfully matches the match rules, and if there is an action rule configured with operation return-permit, route will be permitted and if return-permit is not configured route will be denied. In addition, further operations like modifying route attributes might be executed.

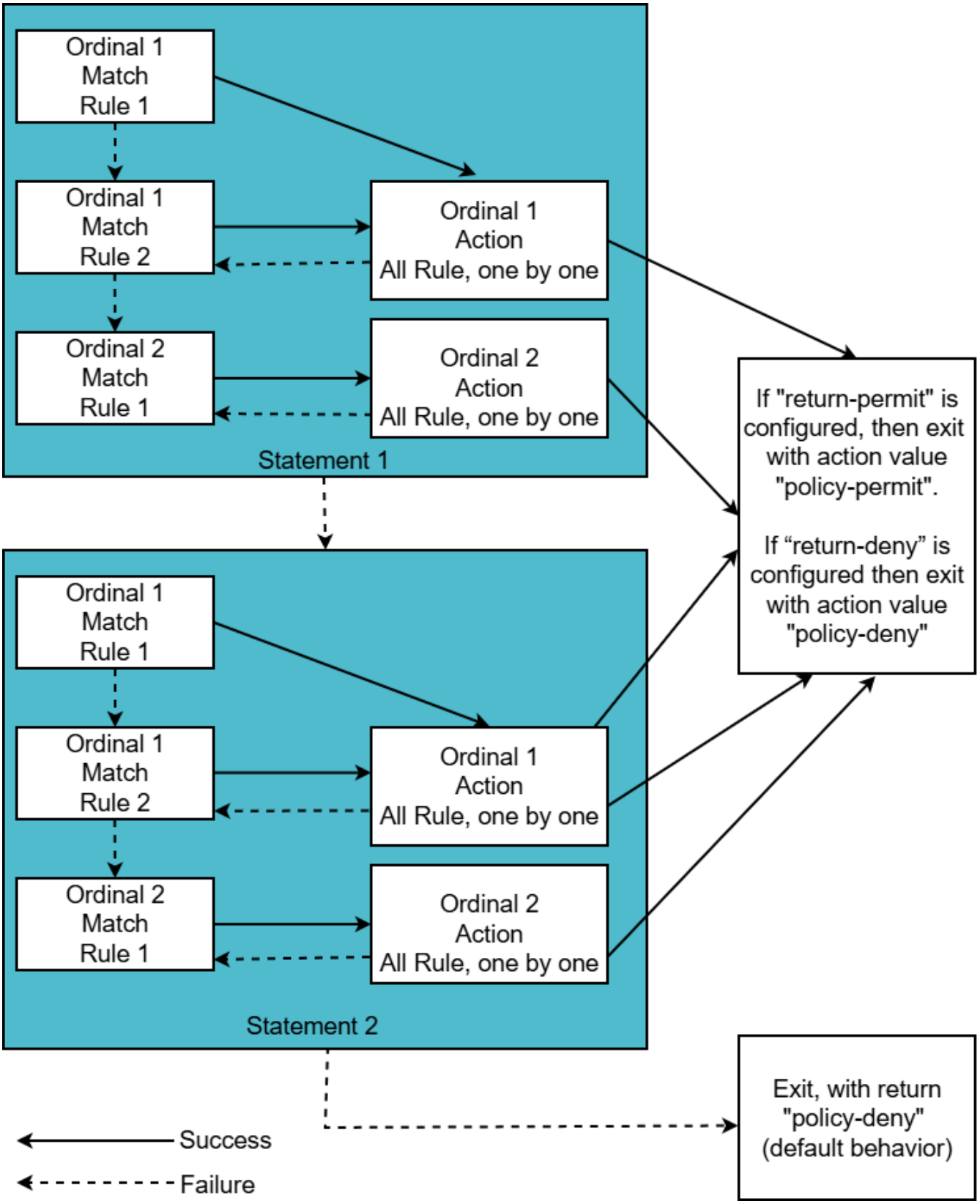
Policy ordinals are executed in the order of the ordinal numbers.

- If an ordinal results in a terminating action like permit or deny, the policy processing is completed for the respective object, and subsequent ordinals will not be processed.
- If an ordinal does not result in a match, the next ordinal is processed.

There might be situations in which a policy configuration is not complete or not valid. In particular, a policy may contain a match or action type not supported by an attachment point. For example, the ipv4-mcast-group type is not supported by BGP or IS-IS. The following list summarizes the behaviour for such invalid scenarios:

- If a policy contains only statements, or only statements and ordinals, but no match and action block, it will deny all.
- If a match or action type is not supported by the attachment point, the policy will ignore the unsupported rule and process only the supported rules. For any ignored rule, the default deny is not impacted. The behaviour will be the same as if the unsupported rule does not exist.

The following diagram shows how ordinals are processed in a policy.




**Match Type**

The Match Type refers to the criteria or conditions used to determine if a particular route or set of routes should be acted upon by the policy. It specifies which attributes of the routes are to be evaluated to see if they meet the policy conditions.

The following diagram shows various match types supported.

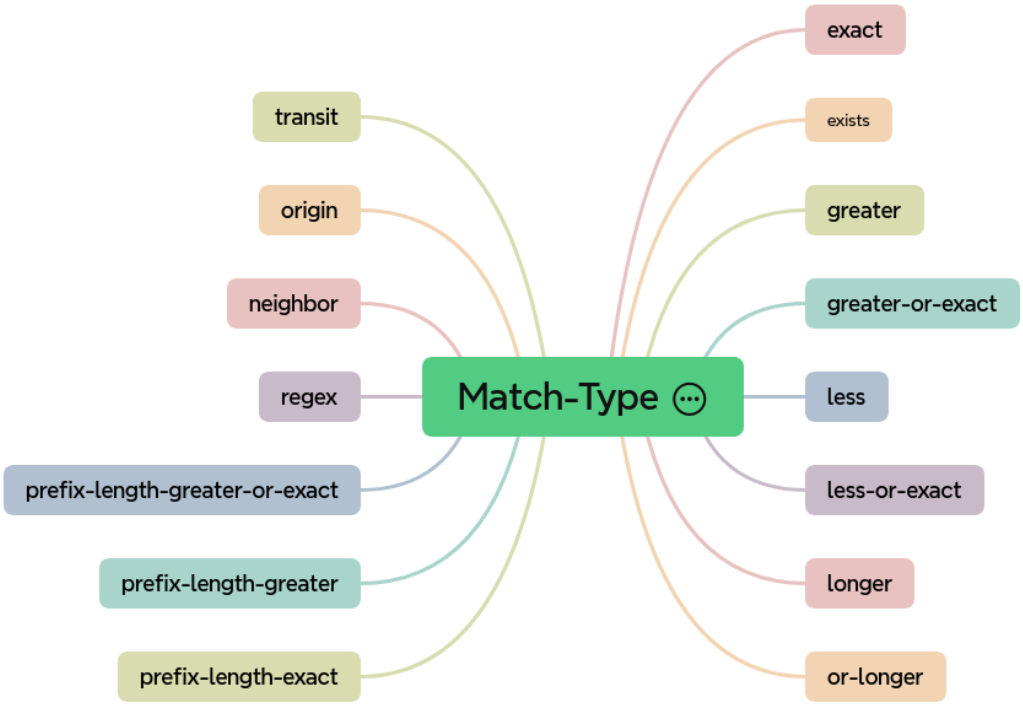



*Figure 38. Match Types*

**Operation Type**

The Operation Type refers to the actions applied to the routes that match the specified criteria or match types within the policy. Once a route matches the conditions set by the match types, the operation type dictates what is to be done with the route.

The following diagram shows various operation types supported.

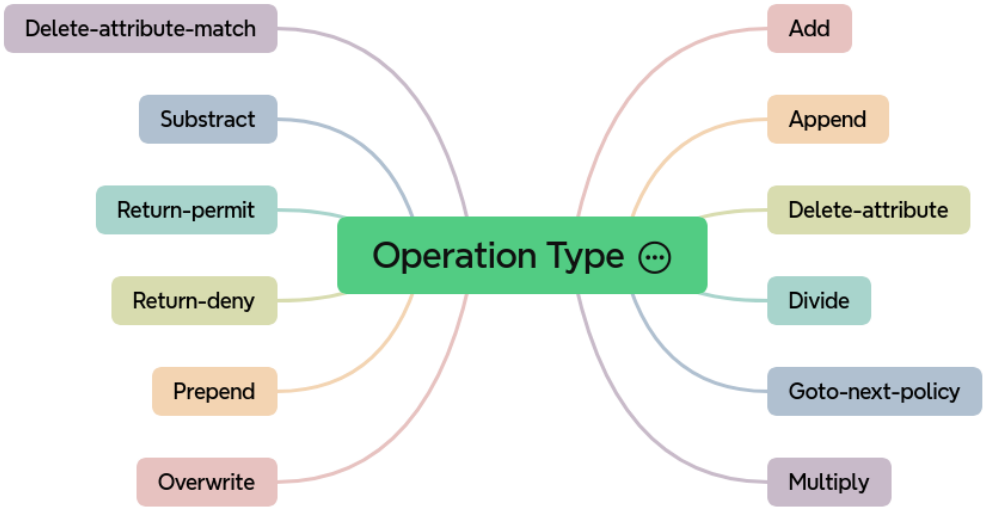



*Figure 39. Operation Types*

**Attribute Type**

The Attribute Type refers to the specific properties of routes that can be matched against or modified within the policy. These attributes are used to define conditions (match types) or actions (operation types) on the routes.

The following diagram shows various attribute types supported.

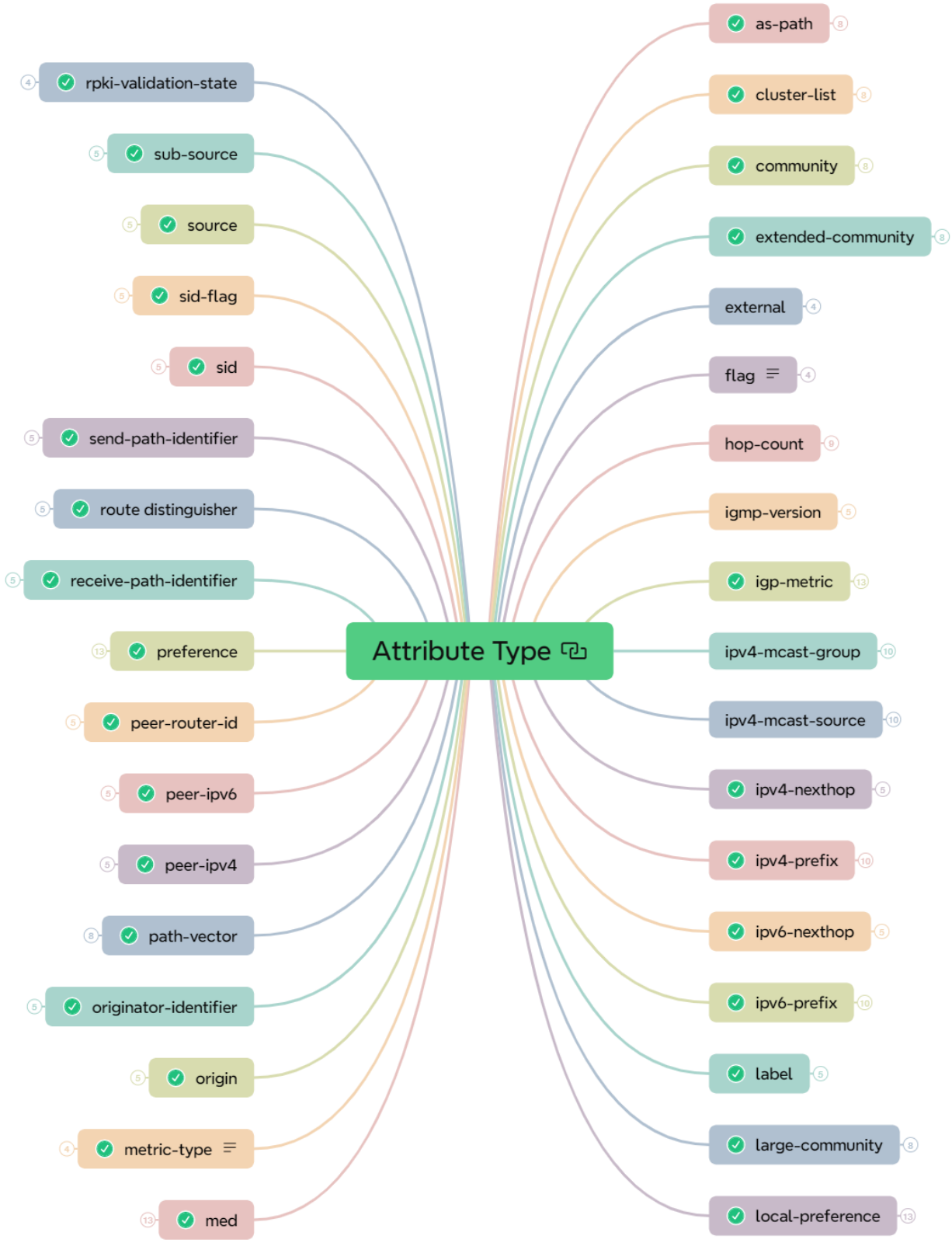



*Figure 40. Attribute Types*

### Policy Chaining

Multiservice edge nodes, such as Broadband Network Gateways (BNG) and peering points, manage routes from various sources. The complexity of policy configurations for these routes can be significant. Certain policies, like filtering private IP addresses and removing private Autonomous System (AS) numbers from the AS path, are frequently reused across different router application

points. Therefore, implementing modularization and enabling the chaining of these policies improve both maintainability and clarity in policy configurations.

Policy chaining is used to modularize monolithic complex routing policies into multiple smaller policies that are executed in a predefined order and are easier to maintain.

Benefits of using policy chaining:

- **Rule Reusability**: Reuse common rules (for example, RFC1918, bogon filtering) across multiple chains, simplifying configuration.
- **Custom Policies**: Create customer-specific policies by combining existing rules for different peering scenarios (transit, upstream, public).
- **Easier Rule Updates**: Simplify updates to general rules affecting multiple policies.
- **Simplify Troubleshooting**: Make troubleshooting easier by breaking down complex configurations into reusable components.

The diagram below shows how a chain of routing policies is evaluated. When Policy Statements 1-3 are chained in configuration, the actual policy evaluation should proceed as shown in the figure below.

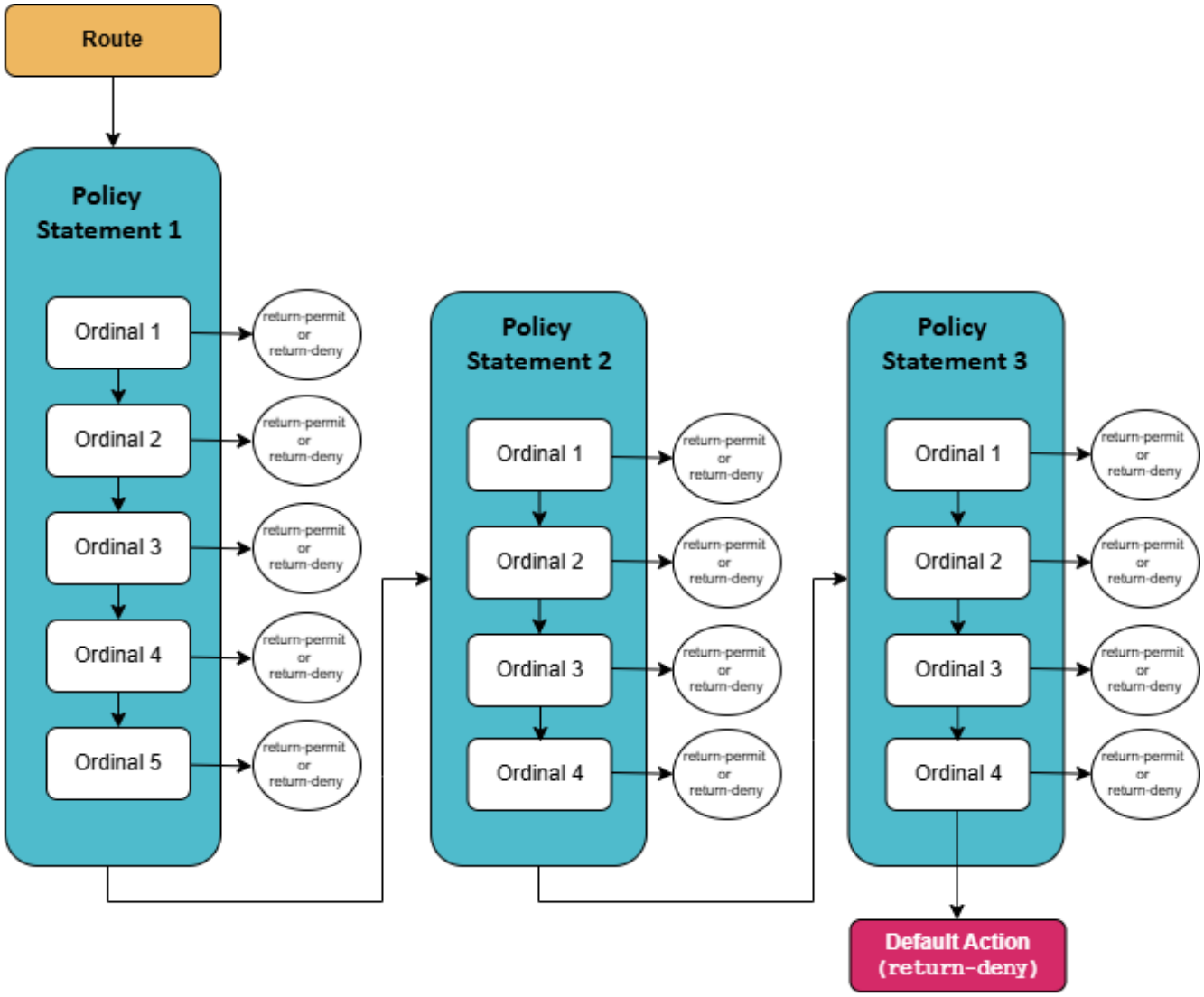



*Figure 41. Policy Chaining Workflow*

As shown in the diagram, the route is evaluated starting with the first ordinal (Ordinal 1) in the first routing policy (Policy Statement 1).

- If there is a match, the specified action is performed. If the action is to return-permit or return-deny the route, that action is performed, and the evaluation of the route ends.
  - If there is no match in the first ordinal (Ordinal 1) of the first routing policy (Policy Statement 1), the evaluation continues by processing the next ordinal (Ordinal 2) and so on.
  - If the route does not match any of the ordinals in the first routing policy (Policy Statement 1), the evaluation continues by processing the ordinals of the second routing policy (Policy Statement 2) and so on.
  - If there are no more routing policies to evaluate, the evaluation ends as the default action is `return-deny`.

#### Support for BGP Community-based Route Leaking Policy

RBFS supports BGP `community`, `extended-community`, and `large-community` as match conditions in route leaking policies. These community attributes can be used to define more flexible route leaking policies.

The table below lists the newly added attributes and match types of route leaking policy.

*Match Types Supported for Community Attributes*

| Attribute Type | Match Types Supported |
|---|---|
| community | regex<br>exact<br>exists |
| extended-community | regex<br>exact<br>exists |
| large-community | regex<br>exact<br>exists |

### Guidelines and Limitations

The following guidelines and limitations apply to policy configuration and usage:

- Policy cannot be attached without configuring policy statement configuration.
- Conditional policy cannot be attached without policy conditional statement configurations.
- The match value list name cannot be configured under a policy statement without policy list configurations.
- If a policy statement is referenced in any attachment point, users cannot delete the policy statement until its references are removed.
- If a policy conditional statement is referenced in any attachment point, users cannot delete the policy conditional statement until its references are removed.

- If a policy list references a policy statement, users cannot delete the policy list until its references are removed.

#### Supported Platforms

Not all features are necessarily supported on each hardware platform. Please refer to the RBFS Platform Guide for the features and the sub-features supported or not supported for each platform.

### 2.9.2. Policy Configuration

#### Configuration Hierarchy

The diagram illustrates the policy configuration hierarchy.

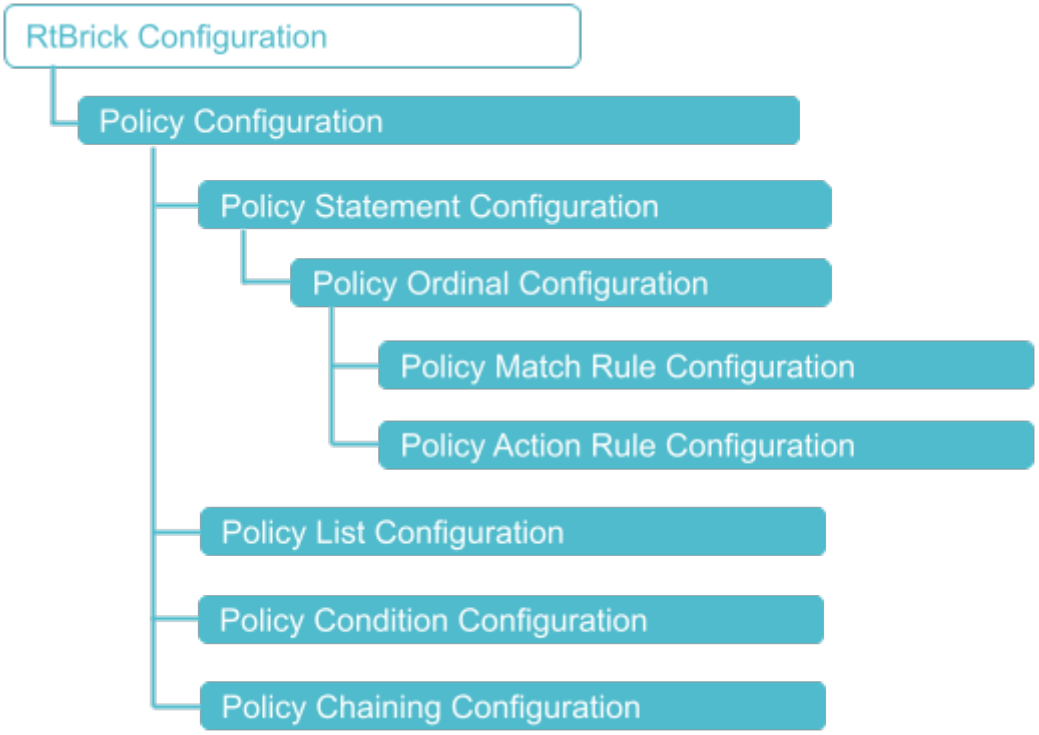



*Figure 42. Policy Configuration Hierarchy*

#### Configuration Syntax and Commands

The following sections describe the policy configuration syntax and commands.

##### Configuring Policy Statements

**Syntax:**

```
set policy statement <policy-name> ordinal <number> <attribute> <value>
```

| Attribute | Description |
|---|---|
| `<policy-name>` | Name of the policy statement. Policy names can contain alphanumeric characters and underscore character. They must not include special characters like hyphen. For example, **BGP-EXPORT** is not supported, whereas **BGP_EXPORT** is supported. A valid name cannot start with a number but it can contain numbers and underscore in the string. The length of the names should not exceed 64 characters. |
| `<number>` | Specifies the ordinal number. |
| `description <text>` | Description of the ordinal. |
| `match <…>` | Match configuration hierarchy. Refer to the Configuring Match Rules section for details on the match rule configuration. |
| match-logic <value> | Specifies the match logic. Supported values are `and` and `or`. Default is `or`. |
| `action <…>` | Action configuration hierarchy. Refer to the Action Rule Configuration section for details on the action rule configuration. |
| `condition <condition-name>` | Optionally, apply a policy condition. Refer to the Configuring Policy Conditions for the policy condition configuration. |

#### Configuring Match Rules

**Syntax:**

**set policy statement** <policy-name> **ordinal** <number> **match rule** <number> <attribute> <value>

| Attribute | Description |
|---|---|
| `rule <number>` | Specifies the match rule number. |
| `type <attribute-type>` | Specifies the attribute type. Refer to the Attribute and Match Types section for details on the supported attribute types. |
| `match-type <match-type>` | Specifies the match type. Refer to the Attribute and Match Types section for details on the supported match types per attribute, and to the Match Types section for descriptions of the match types. |
| `value <value>` | Attribute value. This is the actual value of the attribute to match, for example an IP prefix, a metric, or a community. |

| Attribute | Description |
| --- | --- |
| `value2 <value>` | Attribute value2.<br>While specifying the AS-path, the sub-type "range" allows configuring a range of AS path values. The policy list supports sub-type "range" and ordinal value "value2", which are supported only for AS-path. When sub-type "range" is configured, the "value" (minimum value) and "value2" (maximum value) must be configured. |
| `value-type <value-type>` | Attribute value type.<br>Supported value types are:<br>`discrete` - this is a single value<br>`list` - this is a list of values defined in a policy list<br>`length` - to filter RIB entries based on the length of AS paths |

**Attribute and Match Types**

| Attribute Type | Match Types Supported |
| --- | --- |
| `as-path` | regex<br>exact<br>exists<br>origin<br>neighbor<br>transit |
| `cluster-list` | regex<br>exact<br>exists |
| `community` | regex<br>exact<br>exists |
| `extended-community` | regex<br>exact<br>exists |
| `external` | exact |
| `flag` | exact |
| `hop` | regex<br>exact<br>lesser<br>greater<br>greater-or-exact<br>lesser-or-exact |

| Attribute Type | Match Types Supported |
| --- | --- |
| `igp-metric` | regex<br>exact<br>greater<br>greater-or-exact<br>less<br>less-or-exact |
| `igmp-version` | regex<br>exact |
| `ipv4-mcast-group` | regex<br>exact<br>longer<br>or-longer<br>prefix-length-exact<br>prefix-length-greater<br>prefix-length-greater-or-exact |
| `ipv4-mcast-source` | regex<br>exact<br>longer<br>or-longer<br>prefix-length-exact<br>prefix-length-greater<br>prefix-length-greater-or-exact |
| `ipv4-nexthop` | regex<br>exact |
| `ipv4-prefix` | regex<br>exact<br>longer<br>or-longer<br>prefix-length-exact<br>prefix-length-greater<br>prefix-length-greater-or-exact |
| `ipv6-nexthop` | regex<br>exact |
| `ipv6-prefix` | regex<br>exact<br>longer<br>or-longer<br>prefix-length-exact<br>prefix-length-greater<br>prefix-length-greater-or-exact |
| `label` | regex<br>exact |

| Attribute Type | Match Types Supported |
| --- | --- |
| `large-community` | regex<br>exact<br>exists |
| `local-preference` | regex<br>exact<br>lesser<br>greater<br>greater-or-exact<br>lesser-or-exact |
| `metric-type` | exact |
| `med` | regex<br>exact<br>lesser<br>greater<br>greater-or-exact<br>lesser-or-exact |
| `origin` | regex<br>exact |
| `originator-identifier` | regex<br>exact |
| `path-vector` | regex<br>exact<br>exists |
| `peer-ipv4` | regex<br>exact |
| `peer-ipv6` | regex<br>exact |
| `peer-router-id` | regex<br>exact |
| `preference` | exact<br>lesser<br>greater<br>greater-or-exact<br>lesser-or-exact |
| `receive-path-identifier` | regex<br>exact |
| `route-distinguisher` | regex<br>exact |
| `rpki-validation-state` | exact |

| Attribute Type | Match Types Supported |
| --- | --- |
| `send-path-identifier` | regex<br>exact |
| `sid` | regex<br>exact |
| `source` | regex<br>exact |
| `sub-source` | regex<br>exact |
| `sid-flag` | regex<br>exact |

**Match Types**

| Match Types | Description |
| --- | --- |
| `regex` | An attribute can be matched using a standard Linux egrep regular expression.<br><br>Example: "label": "label-op:push,label:206,bos:1"<br><br>In this example, the label is a 64bit number, which has label value, bos, and operation encoding. A `regex` is used to match the string which is displayed in the table dump, that is, label-op:push,label:206,bos:1 not the 64bit value. The same is applicable to an array type attribute. A `regex` can be written to the string which is visible in the table dump output. For information about Well-known Communities match with Regex, see section "Well-known Communities Regex Match". |
| `exact` | Value configured in the command must be same as application attribute value |
| `exists` | This is applicable only for array type attribute; an exist match is the one where value configured in the command must exist in the application attribute value which is an array. |
| `less` | The application attribute value must be less than the value configured in the command |
| `less-or-exact` | The application attribute value must be less than or exact value configured in the command |
| `greater` | The application attribute value must be greater than the value configured in the command |
| `greater-or-exact` | The application attribute value must be greater than or exact value configured in the command |
| `greater-longer` | The route shares the same most-significant bits (described by prefix-length), and prefix-length is greater than the route’s prefix length |

| Match Types | Description |
|---|---|
| `greater-or-longer` | The route shares the same most-significant bits (described by prefix-length), and prefix-length is equal to or greater than the route's prefix length. |
| `longer` | The route address shares the same most-significant bits as the match prefix (destination-prefix or source-prefix). The number of significant bits is described by the prefix-length component of the match prefix. The match will be performed only for prefixes longer than the one supplied for matching. |
| `or-longer` | The route address shares the same most-significant bits as the match prefix (destination-prefix or the source-prefix). The number of significant bits is described by the prefix-length component of the match prefix. The match is performed for the prefix supplied and for any prefixes that are subnets of it. |
| `neighbor` | The "neighbor" match-type specifically targets the AS number of the neighboring peer in the RIB entry. |
| `transit` | The "transit" match-type matches for AS numbers in rib-entry excluding neighbor and origin. |
| `origin` | The "origin" match-type matches for origin as-path number. |
| `prefix-length-exact` | The application attribute value whose prefix length must be lesser than or exact the value configured in the command |
| `prefix-length-greater` | The application attribute value whose prefix length must be greater than the value configured in the command |
| `prefix-length-greater-or-exact` | The application attribute value whose prefix length must be greater than or exact value configured in the command |

#### Well-known Communities Regex Match

You must specify the value differently to match well-known communities using regex. For example, you can use the following regex pattern to match well-known communities: NO_EXPORT, NO_ADVERTISE, NO_EXPORT_SUBCONFED, and NOOPER.

This pattern works as follows:

NO* matches any string that starts with NO.

This will successfully match the following well-known communities because they start with `NO`.

- NO_EXPORT
- NO_ADVERTISE
- "NO_EXPORT_SUBCONFED"
- "NOPEER"

#### Configuring Action Rules

**Syntax:**

**set policy statement** <policy-name> **ordinal** <number> **action rule** <number> <attribute> <value>

| Attribute | Description |
|---|---|
| `rule <number>` | Specifies the action rule number. |
| `operation <operation-type>` | Specifies the operation type. Refer to Operation Types section for details on the supported operations, and to the Attribute Types and Supported Operations section for details on the operations per attribute. |
| `type <attribute-type>` | Specifies the attribute type. Refer to section Attribute Types and Supported Operations section for details on the supported attribute types. |
| `value <value>` | Specifies the operation value. |

**Operation Types**

| Operation Type | Description |
|---|---|
| `add` | The application attribute value will be added with the value configured in the command |
| `append` | The application attribute value will be appended with the value configured in the command |
| `delete-attribute` | Deletes the attribute from the route. It clears all the information for that specific attribute in the object. |
| `delete-attribute-match` | Deletes the specific value of an attribute from the route. It is mandatory to define match rules, before configuring the `delete-attribute-match` action rule. |
| `divide` | The application attribute value will be divided with the value configured in the command |
| `goto-next-policy` | (Optional). By default, continue with the next policy in the policy chain. |
| `multiply` | The application attribute value will be multiplied with the value configured in the command |
| `overwrite` | The application attribute value will be overwritten with the value configured in the command |
| `prepend` | The application attribute value will be prefixed with the value configured in the command |
| `return-deny` | Stops policy execution and returns result as `deny` (resulting route/BDS object to be denied) |
| `return-permit` | Stops policy execution and return result as permit (resulting route/BDS object to be permitted) |

| Operation Type | Description |
| --- | --- |
| `subtract` | The application attribute value will be subtracted with the value configured in the command. If the result of the subtraction results in a number less than 0, the value "0" is used. |

**Attribute Types and Supported Operations**

| Attribute Type | Operation Types Supported |
| --- | --- |
| `as-path` | append<br>prepend<br>overwrite |
| `cluster-list` | append<br>prepend<br>overwrite |
| `community` | append<br>prepend<br>overwrite |
| `extended-community` | append<br>prepend<br>overwrite |
| `external` | overwrite |
| `flag` | overwrite |
| `hop-count` | overwrite |
| `igp-metric` | add<br>subtract<br>multiply<br>divide<br>overwrite |
| `igmp-version` | overwrite |
| `ipv4-mcast-group` | overwrite |
| `ipv4-mcast-source` | overwrite |
| `ipv4-nexthop` | overwrite |
| `ipv4-prefix` | overwrite |
| `ipv6-nexthop` | overwrite |
| `ipv6-prefix` | overwrite |
| `label` | overwrite |
| `large-community` | append<br>prepend<br>overwrite |

| Attribute Type | Operation Types Supported |
|---|---|
| `local-preference` | add<br>substract<br>multiply<br>divide<br>overwrite |
| `med` | add<br>substract<br>multiply<br>divide<br>overwrite |
| `metric-type` | overwrite |
| `origin` | overwrite |
| `originator-identifier` | overwrite |
| `path-vector` | append<br>prepend<br>overwrite |
| `peer-ipv4` | overwrite |
| `peer-ipv6` | overwrite |
| `peer-router-id` | overwrite |
| `preference` | add<br>substract<br>multiply<br>divide<br>overwrite |
| `receive-path-identifier` | overwrite |
| `route-distinguisher` | overwrite |
| `rpki-validation-state` | overwrite |
| `send-path-identifier` | overwrite |
| `sid` | overwrite |
| `sid-flag` | overwrite |
| `source` | overwrite |
| `sub-source` | overwrite |

Example 1: Policy statement configuration

```
{
    "rtbrick-config:policy": {
      "statement": [
        {
```

```
        "name": "EXPORT_POLICY1",
        "ordinal": [
          {
            "ordinal": 10,
            "description": "Add BGP community to direct routes",
            "match-logic": "and",
            "match": {
              "rule": [
                {
                  "rule": 1,
                  "type": "ipv6-prefix",
                  "value-type": "discrete",
                  "match-type": "or-longer",
                  "value": "2001:db8:0:60::/32"
                },
                {
                  "rule": 2,
                  "type": "source",
                  "value-type": "discrete",
                  "match-type": "exact",
                  "value": "direct"
                }
              ]
            },
            "action": {
              "rule": [
                {
                  "rule": 1,
                  "type": "community",
                  "operation": "append",
                  "value": "100:1"
                },
                {
                  "rule": 2,
                  "operation": "return-permit"
                }
              ]
            }
          },
          {
            "ordinal": 20,
            "description": "Allow any other route",
            "action": {
              "rule": [
                {
                  "rule": 1,
                  "operation": "return-permit"
                }
              ]
            }
          }
```

```
          ]
        }
      ]
    }
  }
```

Example 2: Policy statement configuration for RPKI Validation State Overwrite

```
{
  "rtbrick-config:policy": {
    "statement": [
      {
        "name": "S1",
        "ordinal": [
          {
            "ordinal": 1,
            "match-logic": "and",
            "match": {
              "rule": [
                {
                  "rule": 1,
                  "type": "rpki-validation-state",
                  "value-type": "discrete",
                  "match-type": "exact",
                  "value": [
                    "invalid"
                    ]
                },
                {
                  "rule": 2,
                  "type": "ipv4-prefix",
                  "value-type": "discrete",
                  "match-type": "exact",
                  "value": [
                    "192.168.0.11/32"
                    ]
                }
              ]
            },
            "action": {
              "rule": [
                {
                  "rule": 1,
                  "type": "rpki-validation-state",
                  "operation": "overwrite",
                  "value": [
                    "unknown"
                    ]
                },
                {
```

```
                "rule": 2,
                "operation": "return-permit"
              }
            ]
          }
        }
      ]
    }
  ]
}
}
```

#### Configuring Policy Lists

**Syntax:**

**set policy list** <list-name> <list-type> **ordinal** <ordinal-number> **value** <value>

| Attribute | Description |
|---|---|
| `<list-name>` | Name of the policy list |
| `<list-type>` | Type of the policy list. The following types of lists are supported:<br><br>- as-path<br>- cluster-list<br>- community<br>- route-distinguisher<br>- extended-community<br>- ipv4-address<br>- ipv4-mcast-group<br>- ipv4-prefix<br>- ipv6-address<br>- ipv6-prefix<br>- l2-prefix<br>- large-community<br>- mpls-label<br>- source<br>- sub-source |
| `<ordinal-number>` | The number of the list entry. |
| `<value>` | The value of the list entry, for example an IP prefix, or a community. |

Example: Policy list configuration

```
{
```

```
    "rtbrick-config:policy": {
      "list": [
        {
          "name": "PREFIX_LIST1",
          "type": "ipv6-prefix",
          "ordinal": [
            {
              "ordinal": 1,
              "value": "2001:db8:0:10::/32"
            },
            {
              "ordinal": 2,
              "value": "2001:db8:0:25::/32"
            },
            {
              "ordinal": 3,
              "value": "2001:db8:0:30::/32"
            }
          ]
        }
      ]
    }
}
```

Example: Optimising AS-path Matching

The "neighbor" match-type specifically targets the AS number of the neighboring peer in the RIB entry. For example, the following configuration matches RIB entries whose neighbor AS number is in the "as_neighbor_list", such as 100, 200, or 300.

```
set policy list as_neighbor_list as-path
set policy list as_neighbor_list as-path ordinal 1
set policy list as_neighbor_list as-path ordinal 1 value 100
set policy list as_neighbor_list as-path ordinal 2
set policy list as_neighbor_list as-path ordinal 2 value 200
set policy list as_neighbor_list as-path ordinal 3
set policy list as_neighbor_list as-path ordinal 3 value 300
set policy statement IMPORT_POLICY
set policy statement IMPORT_POLICY ordinal 1
set policy statement IMPORT_POLICY ordinal 1 match
set policy statement IMPORT_POLICY ordinal 1 match rule 1
set policy statement IMPORT_POLICY ordinal 1 match rule 1 type as-path
set policy statement IMPORT_POLICY ordinal 1 match rule 1 value-type list
set policy statement IMPORT_POLICY ordinal 1 match rule 1 match-type neighbor
set policy statement IMPORT_POLICY ordinal 1 match rule 1 value as_neighbor_list
set policy statement IMPORT_POLICY ordinal 1 action
set policy statement IMPORT_POLICY ordinal 1 action rule 1
set policy statement IMPORT_POLICY ordinal 1 action rule 1 operation return-deny
```

The "origin" match-type matches for origin as-path number. For example, the following configuration matches RIB entries whose origin AS number is in the "as_origin_list".

```
set policy list as_origin_list as-path
set policy list as_origin_list as-path ordinal 1
set policy list as_origin_list as-path ordinal 1 value 125
set policy list as_origin_list as-path ordinal 2
set policy list as_origin_list as-path ordinal 2 value 225
set policy list as_origin_list as-path ordinal 3
set policy list as_origin_list as-path ordinal 3 value 325
set policy statement IMPORT_POLICY ordinal 2
set policy statement IMPORT_POLICY ordinal 2 match
set policy statement IMPORT_POLICY ordinal 2 match rule 1
set policy statement IMPORT_POLICY ordinal 2 match rule 1 type as-path
set policy statement IMPORT_POLICY ordinal 2 match rule 1 value-type list
set policy statement IMPORT_POLICY ordinal 2 match rule 1 match-type origin
set policy statement IMPORT_POLICY ordinal 2 match rule 1 value as_origin_list
set policy statement IMPORT_POLICY ordinal 2 action
set policy statement IMPORT_POLICY ordinal 2 action rule 1
set policy statement IMPORT_POLICY ordinal 2 action rule 1 operation return-deny
```

The "transit" match-type matches for AS numbers in rib-entry excluding neighbor and origin.

```
set policy list as_transit_list as-path
set policy list as_transit_list as-path ordinal 1
set policy list as_transit_list as-path ordinal 1 value 150
set policy list as_transit_list as-path ordinal 2
set policy list as_transit_list as-path ordinal 2 value 250
set policy list as_transit_list as-path ordinal 3
set policy list as_transit_list as-path ordinal 3 value 350
set policy statement IMPORT_POLICY ordinal 3
set policy statement IMPORT_POLICY ordinal 3 match
set policy statement IMPORT_POLICY ordinal 3 match rule 1
set policy statement IMPORT_POLICY ordinal 3 match rule 1 type as-path
set policy statement IMPORT_POLICY ordinal 3 match rule 1 value-type list
set policy statement IMPORT_POLICY ordinal 3 match rule 1 match-type transit
set policy statement IMPORT_POLICY ordinal 3 match rule 1 value as_transit_list
set policy statement IMPORT_POLICY ordinal 3 action
set policy statement IMPORT_POLICY ordinal 3 action rule 1
set policy statement IMPORT_POLICY ordinal 3 action rule 1 operation return-deny
```

The sub-type "range" allows configuring a range of AS path values. The policy list supports sub-type "range" and ordinal value "value2", which are supported only for AS-path. When sub-type "range" is configured, the "value" (minimum value) and "value2" (maximum value) must be configured.

```
set policy list BOGON_ASNS as-path
set policy list BOGON_ASNS as-path sub-type range
set policy list BOGON_ASNS as-path ordinal 1
```

```
set policy list BOGON_ASNS as-path ordinal 1 value 0
set policy list BOGON_ASNS as-path ordinal 1 value2 0
set policy list BOGON_ASNS as-path ordinal 2
set policy list BOGON_ASNS as-path ordinal 2 value 23456
set policy list BOGON_ASNS as-path ordinal 2 value2 23456
set policy list BOGON_ASNS as-path ordinal 3
set policy list BOGON_ASNS as-path ordinal 3 value 64496
set policy list BOGON_ASNS as-path ordinal 3 value2 64511
set policy list BOGON_ASNS as-path ordinal 4
set policy list BOGON_ASNS as-path ordinal 4 value 65536
set policy list BOGON_ASNS as-path ordinal 4 value2 65551
set policy list BOGON_ASNS as-path ordinal 5
set policy list BOGON_ASNS as-path ordinal 5 value 4200000000
set policy list BOGON_ASNS as-path ordinal 5 value2 4294967294
set policy list BOGON_ASNS as-path ordinal 6
set policy list BOGON_ASNS as-path ordinal 6 value 65535
set policy list BOGON_ASNS as-path ordinal 6 value2 65535
set policy list BOGON_ASNS as-path ordinal 7
set policy list BOGON_ASNS as-path ordinal 7 value 4294967294
set policy list BOGON_ASNS as-path ordinal 7 value2 4294967294
set policy list BOGON_ASNS as-path ordinal 8
set policy list BOGON_ASNS as-path ordinal 8 value 65552
set policy list BOGON_ASNS as-path ordinal 8 value2 131071

set policy statement BOGON_ASN_FILTERING
set policy statement BOGON_ASN_FILTERING ordinal 1
set policy statement BOGON_ASN_FILTERING ordinal 1 match
set policy statement BOGON_ASN_FILTERING ordinal 1 match rule 1
set policy statement BOGON_ASN_FILTERING ordinal 1 match rule 1 type as-path
set policy statement BOGON_ASN_FILTERING ordinal 1 match rule 1 value-type list
set policy statement BOGON_ASN_FILTERING ordinal 1 match rule 1 match-type exists
set policy statement BOGON_ASN_FILTERING ordinal 1 match rule 1 value BOGON_ASNS
set policy statement BOGON_ASN_FILTERING ordinal 1 action
set policy statement BOGON_ASN_FILTERING ordinal 1 action rule 1
set policy statement BOGON_ASN_FILTERING ordinal 1 action rule 1 operation return-deny
```

Also, network operators may need to match AS-path length to detect bogus announcements. For example, drop all AS-paths with lengths greater than 100. To support this, a new value type, "length" has been added, which supports the match type "greater-or-exact."

```
set policy statement CHECK_LENGTH
set policy statement CHECK_LENGTH ordinal 1
set policy statement CHECK_LENGTH ordinal 1 match
set policy statement CHECK_LENGTH ordinal 1 match rule 1
set policy statement CHECK_LENGTH ordinal 1 match rule 1 type as-path
set policy statement CHECK_LENGTH ordinal 1 match rule 1 value-type length
set policy statement CHECK_LENGTH ordinal 1 match rule 1 match-type greater-or-exact
set policy statement CHECK_LENGTH ordinal 1 match rule 1 value 10
set policy statement CHECK_LENGTH ordinal 1 action
set policy statement CHECK_LENGTH ordinal 1 action rule 1
```

```
set policy statement CHECK_LENGTH ordinal 1 action rule 1 operation return-permit
```

Example: Configuring Layer 2 prefix matches in policy configurations

The configuration example below defines an L2 prefix policy with two rules, where the first rule matches an exact discrete value of a specific L2 prefix and permits it, while the second rule denies all other traffic.

```
set policy statement L2_prefix_import_policy ordinal 1 match rule 1 type l2-prefix
set policy statement L2_prefix_import_policy ordinal 1 match rule 1 value-type
discrete
set policy statement L2_prefix_import_policy ordinal 1 match rule 1 match-type exact
set policy statement L2_prefix_import_policy ordinal 1 match rule 1 value
00.00.00.00.00.00.00.00.00.00:301/112
set policy statement L2_prefix_import_policy ordinal 1 action rule 2 operation return-
permit
set policy statement L2_prefix_import_policy ordinal 2 action rule 1 operation return-
deny
```

```
{
  "rtbrick-config:statement": [
    {
      "name": "L2_prefix_policy",
      "ordinal": [
        {
          "ordinal": 1,
          "match": {
            "rule": [
              {
                "rule": 1,
                "type": "l2-prefix",
                "value-type": "discrete",
                "match-type": "exact",
                "value": [
                  "00.00.00.00.00.00.00.00.00.00:301/112"
                  ]
              }
            ]
          },
          "action": {
            "rule": [
              {
                "rule": 2,
                "operation": "return-permit"
              }
            ]
          }
        },
```

```
        {
          "ordinal": 2,
          "action": {
            "rule": [
              {
                "rule": 1,
                "operation": "return-deny"
              }
            ]
          }
        }
      ]
    }
  ]
}
```

**Attaching a Policy:**

The example below shows how to attach a policy statement with an L2 prefix match rule to a BGP peer group (import/export).

```
set instance default protocol bgp peer-group PEER-GROUP1 address-family l2vpn evpn
policy import L2_prefix_import_policy
set instance default protocol bgp peer-group PEER-GROUP1 address-family l2vpn evpn
policy export L2_prefix_export_policy
```

#### Configuring Policy Conditions

Policy conditions refer to certain states of the system represented in BDS tables. You need to specify the table, the daemon (BD) that needs to resolve the condition, and if applicable, the attributes used to define the condition.

There are two types of conditions:

- Match on certain attributes in BDS table objects.
- Match on the number of objects in a BDS table.

#### Configuring Table Match

**Syntax:**

**set policy condition** <condition-name> **table** <attribute> <value>

| Attribute | Description |
|---|---|
| `<condition-name>` | Name of the policy condition |
| `name <table-name>` | Name of the BDS table |

| Attribute | Description |
|---|---|
| `bd <bd-name>` | Name of the BD which resolves the condition. Currently supported BDs are: ifmd, ribd, mribd, pppoed, subscriberd, ipoed, l2tpd, pimd, igmp.iod, isis.iod, ospf.appd ospf.iod |
| `count <number>` | Optionally, match on the number of objects in a table |
| `match <type>` | Type of match for an object count. Supported match types are:<br>equal<br>greater<br>greater-or-equal<br>less<br>less-or-equal |

#### Configuring Attribute Match

You can configure attributes to define a condition. The attributes refer to objects in the table configured in the section above. Please note:

- You can match on multiple attributes.
- In order to identify matching objects, you need to specify all attributes which are primary keys in the table.
- Attribute configuration is not required for conditions matching on the number of table objects using the `count` option.

**Syntax:**

**set policy condition** <condition-name> **attribute** <name> <attribute> <value>

| Attribute | Description |
|---|---|
| `<condition-name>` | Name of the policy condition |
| `attribute <name>` | Name of the attribute in the BDS table |
| `value <value>` | Value of the attribute to match |
| `match <type>` | Type of the match. Supported match types are:<br>equal<br>exists<br>greater<br>greater-or-equal<br>less<br>less-or-equal<br>regex |

Example: Policy condition configuration

```
{
```

```
    "rtbrick-config:policy": {
      "condition": [
        {
          "condition_name": "precheck",
          "table": {
            "name": "global.instance",
            "bd": "bgp.iod.1"
          },
          "attribute": [
            {
              "name": "instance_name",
              "match": "equal",
              "value": "default"
            }
          ]
        }
      ]
    }
  }
```

#### Attaching Policies

Once a policy has been created, they need to be applied to an application like a routing protocol to take effect.

#### BGP Attachment Points

- Instance import/export
- BGP peer group import/export
- BGP redistribution

For attaching policies to the BGP protocol, please refer to the RBFS BGP User Guide.

#### IS-IS Attachment Points

- IS-IS redistribution

For attaching policies to the IS-IS protocol, please refer to the IS-IS User Guide.

#### OSPFv2 Attachment Points

- OSPFv2 redistribution

For attaching policies to the OSPFv2/v3 protocol, please refer to the OSPFv2 User Guide.

#### LDP Attachment Points

- LDP redistribution#

For attaching policies to the LDP protocol, please refer to the LDP User Guide.

#### IGMP Attachment Points

- IGMP group filtering
- IGMP SSM-mapping

For attaching policies to the IGMP protocol, please refer to the RBFS IP Multicast Routing Configuration Guide.

#### Policy Chaining Configuration

The following configuration demonstrates how policy chaining is applied for the PEERING_V4 BGP peer group. For IPv4 unicast address family, incoming routes will be evaluated and imported only if they meet the criteria defined in the associated import policies. These policies provide layered route filtering, ensuring that only valid and desired routes are accepted by the peer.

The configuration applies a series of import policies for the IPv4 unicast address family. These policies are chained in sequence, each performing specific tasks such as:

- Filtering bogon IPs (FILTER_BOGONS_V4).
- Preventing the import of the peer's own IP ranges (FILTER_OWN_V4).
- Scrubbing unwanted community tags (SCRUB_COMMUNITIES).

The following example shows the policy chaining configuration for the peer-group "PEERING_V4."

```
set instance default protocol bgp peer-group PEERING_V4
set instance default protocol bgp peer-group PEERING_V4 remote-as 65536
set instance default protocol bgp peer-group PEERING_V4 address-family ipv4 labeled-
unicast
set instance default protocol bgp peer-group PEERING_V4 address-family ipv4 unicast
set instance default protocol bgp peer-group PEERING_V4 address-family ipv4 unicast
policy import FILTER_BOGONS_V4
set instance default protocol bgp peer-group PEERING_V4 address-family ipv4 unicast
policy import FILTER_OWN_V4
set instance default protocol bgp peer-group PEERING_V4 address-family ipv4 unicast
policy import SCRUB_COMMUNITIES
```

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol bgp peer-group
PEERING_V4
{
  "rtbrick-config:peer-group": [
    {
      "pg-name": "PEERING_V4",
      "remote-as": 65536,
      "address-family": [
        {
          "afi": "ipv4",
          "safi": "labeled-unicast"
        },
```

```
        {
          "afi": "ipv4",
          "safi": "unicast",
          "policy": {
            "import": [
              "FILTER_BOGONS_V4",
              "FILTER_OWN_V4",
              "SCRUB_COMMUNITIES"              ]
          }
        },
        {
          "afi": "ipv6",
          "safi": "labeled-unicast"
        },
        {
          "afi": "ipv6",
          "safi": "unicast"
        }
      ]
    }
  ]
}
```

**Policy Statement Configurations for Policy Chaining**

The following configurations provide a detailed representation of the policy statements utilized in the aforementioned policy chaining example.

The example below shows the policy definitions for the FILTER_BOGONS_V4 policy statement.

```
set policy statement FILTER_BOGONS_V4
set policy statement FILTER_BOGONS_V4 ordinal 1
set policy statement FILTER_BOGONS_V4 ordinal 1 description FILTER_BOGONS
set policy statement FILTER_BOGONS_V4 ordinal 1 match-logic or
set policy statement FILTER_BOGONS_V4 ordinal 1 match
set policy statement FILTER_BOGONS_V4 ordinal 1 match rule 1
set policy statement FILTER_BOGONS_V4 ordinal 1 match rule 1 type ipv4-prefix
set policy statement FILTER_BOGONS_V4 ordinal 1 match rule 1 value-type list
set policy statement FILTER_BOGONS_V4 ordinal 1 match rule 1 match-type or-longer
set policy statement FILTER_BOGONS_V4 ordinal 1 match rule 1 value FILTER_BOGONS
set policy statement FILTER_BOGONS_V4 ordinal 1 match rule 2
set policy statement FILTER_BOGONS_V4 ordinal 1 match rule 2 type ipv4-prefix
set policy statement FILTER_BOGONS_V4 ordinal 1 match rule 2 value-type discrete
set policy statement FILTER_BOGONS_V4 ordinal 1 match rule 2 match-type regex
set policy statement FILTER_BOGONS_V4 ordinal 1 match rule 2 value 0.0.0.0/(2[5-
9]|3[0-1])
set policy statement FILTER_BOGONS_V4 ordinal 1 action
set policy statement FILTER_BOGONS_V4 ordinal 1 action rule 1
set policy statement FILTER_BOGONS_V4 ordinal 1 action rule 1 operation return-deny
set policy statement FILTER_BOGONS_V4 ordinal 2
```

```
set policy statement FILTER_BOGONS_V4 ordinal 2 description FILTER_IXP
set policy statement FILTER_BOGONS_V4 ordinal 2 match
set policy statement FILTER_BOGONS_V4 ordinal 2 match rule 1
set policy statement FILTER_BOGONS_V4 ordinal 2 match rule 1 type ipv4-prefix
set policy statement FILTER_BOGONS_V4 ordinal 2 match rule 1 value-type list
set policy statement FILTER_BOGONS_V4 ordinal 2 match rule 1 match-type or-longer
set policy statement FILTER_BOGONS_V4 ordinal 2 match rule 1 value
EXAMPLE_PEERING_LANS_V4
set policy statement FILTER_BOGONS_V4 ordinal 2 action
set policy statement FILTER_BOGONS_V4 ordinal 2 action rule 1
set policy statement FILTER_BOGONS_V4 ordinal 2 action rule 1 operation return-deny
```

The example below shows the configurations of the 'FILTER_BOGONS' policy list used in the 'FILTER_BOGONS_V4' policy statement.

```
supervisor@S1-STD-2-2004>bm04-tst.fsn.rtbrick.net: op> show config set policy list
FILTER_BOGONS
set policy list FILTER_BOGONS ipv4-prefix
set policy list FILTER_BOGONS ipv4-prefix ordinal 1
set policy list FILTER_BOGONS ipv4-prefix ordinal 1 value 0.0.0.0/8
set policy list FILTER_BOGONS ipv4-prefix ordinal 2
set policy list FILTER_BOGONS ipv4-prefix ordinal 2 value 10.0.0.0/8
set policy list FILTER_BOGONS ipv4-prefix ordinal 3
set policy list FILTER_BOGONS ipv4-prefix ordinal 3 value 100.64.0.0/10
set policy list FILTER_BOGONS ipv4-prefix ordinal 4
set policy list FILTER_BOGONS ipv4-prefix ordinal 4 value 127.0.0.0/8
```

The example below shows the configurations of the 'EXAMPLE_PEERING_LANS_V4' policy list used in the 'FILTER_BOGONS_V4' policy statement.

```
set policy list EXAMPLE_PEERING_LANS_V4 ipv4-prefix
set policy list EXAMPLE_PEERING_LANS_V4 ipv4-prefix ordinal 1
set policy list EXAMPLE_PEERING_LANS_V4 ipv4-prefix ordinal 1 value 198.51.100.0/24
set policy list EXAMPLE_PEERING_LANS_V4 ipv4-prefix ordinal 2
set policy list EXAMPLE_PEERING_LANS_V4 ipv4-prefix ordinal 2 value 198.51.101.0/24
set policy list EXAMPLE_PEERING_LANS_V4 ipv4-prefix ordinal 3
set policy list EXAMPLE_PEERING_LANS_V4 ipv4-prefix ordinal 3 value 198.51.102.0/24
set policy list EXAMPLE_PEERING_LANS_V4 ipv4-prefix ordinal 4
set policy list EXAMPLE_PEERING_LANS_V4 ipv4-prefix ordinal 4 value 198.51.104.0/23
```

The example below shows the policy definitions for the FILTER_OWN_V4 policy statement.

```
set policy statement FILTER_OWN_V4
set policy statement FILTER_OWN_V4 ordinal 1
set policy statement FILTER_OWN_V4 ordinal 1 match
set policy statement FILTER_OWN_V4 ordinal 1 match rule 1
set policy statement FILTER_OWN_V4 ordinal 1 match rule 1 type ipv4-prefix
```

```
set policy statement FILTER_OWN_V4 ordinal 1 match rule 1 value-type list
set policy statement FILTER_OWN_V4 ordinal 1 match rule 1 match-type or-longer
set policy statement FILTER_OWN_V4 ordinal 1 match rule 1 value MYPREFIXES_V4
set policy statement FILTER_OWN_V4 ordinal 1 action
set policy statement FILTER_OWN_V4 ordinal 1 action rule 1
set policy statement FILTER_OWN_V4 ordinal 1 action rule 1 operation return-deny
```

The example below shows the configurations of the 'MYPREFIXES_V4' policy list used in the 'FILTER_OWN_V4' policy statement.

```
set policy list MYPREFIXES_V4 ipv4-prefix
set policy list MYPREFIXES_V4 ipv4-prefix ordinal 1
set policy list MYPREFIXES_V4 ipv4-prefix ordinal 1 value 192.0.2.0/24
set policy list MYPREFIXES_V4 ipv4-prefix ordinal 2
set policy list MYPREFIXES_V4 ipv4-prefix ordinal 2 value 198.51.100.0/24
set policy list MYPREFIXES_V4 ipv4-prefix ordinal 3
set policy list MYPREFIXES_V4 ipv4-prefix ordinal 3 value 203.0.113.0/24
```

The example below shows the policy definitions for the SCRUB_COMMUNITIES policy statement.

```
set policy statement SCRUB_COMMUNITIES
set policy statement SCRUB_COMMUNITIES ordinal 1
set policy statement SCRUB_COMMUNITIES ordinal 1 match
set policy statement SCRUB_COMMUNITIES ordinal 1 match rule 1
set policy statement SCRUB_COMMUNITIES ordinal 1 match rule 1 type community
set policy statement SCRUB_COMMUNITIES ordinal 1 match rule 1 value-type discrete
set policy statement SCRUB_COMMUNITIES ordinal 1 match rule 1 match-type regex
set policy statement SCRUB_COMMUNITIES ordinal 1 match rule 1 value 64496:*
set policy statement SCRUB_COMMUNITIES ordinal 1 action
set policy statement SCRUB_COMMUNITIES ordinal 1 action rule 1
set policy statement SCRUB_COMMUNITIES ordinal 1 action rule 1 type community
set policy statement SCRUB_COMMUNITIES ordinal 1 action rule 1 operation delete-
attribute-match
set policy statement SCRUB_COMMUNITIES ordinal 1 action rule 2 operation return-permit
```

**Sample Configurations**

Example 1: BGP export policy referencing a policy list

```
supervisor@leaf1: cfg> show config policy
{
  "rtbrick-config:policy": {
    "list": [
      {
        "name": "PREFIX_LIST2",
        "type": "ipv6-prefix",
        "ordinal": [
```

```
          {
            "ordinal": 1,
            "value": "2001:db8:0:60::/32"
          },
          {
            "ordinal": 2,
            "value": "2001:db8:0:80::/64"
          },
          {
            "ordinal": 3,
            "value": "2001:db8:0:110::/64 "
          }
        ]
      }
    ],
    "statement": [
      {
        "name": "EXPORT_POLICY2",
        "ordinal": [
          {
            "ordinal": 10,
            "description": "Add community to direct routes",
            "match": {
              "rule": [
                {
                  "rule": 1,
                  "type": "source",
                  "value-type": "discrete",
                  "match-type": "exact",
                  "value": "direct"
                }
              ]
            },
            "action": {
              "rule": [
                {
                  "rule": 1,
                  "type": "community",
                  "operation": "append",
                  "value": "100:1"
                },
                {
                  "rule": 2,
                  "operation": "return-permit"
                }
              ]
            }
          },
          {
            "ordinal": 20,
            "description": "Allow list of routes",
```

```
            "match": {
              "rule": [
                {
                  "rule": 1,
                  "type": "ipv6-prefix",
                  "value-type": "list",
                  "match-type": "or-longer",
                  "value": "PREFIX_LIST2"
                }
              ]
            },
            "action": {
              "rule": [
                {
                  "rule": 1,
                  "operation": "return-permit"
                }
              ]
            }
          },
          {
            "ordinal": 30,
            "description": "Deny any other route",
            "action": {
              "rule": [
                {
                  "rule": 1,
                  "operation": "return-deny"
                }
              ]
            }
          }
        ]
      }
    ]
  }
}
```

```
supervisor@leaf1: cfg> show config instance default protocol bgp peer-group spine
address-family ipv6 unicast
{
  "rtbrick-config:address-family": [
    {
      "afi": "ipv6",
      "safi": "unicast",
      "policy": {
        "export": "EXPORT_POLICY2"
      }
    }
  ]
```

```
}
```

Example 2: IGMP filter policy

```
supervisor@leaf1: cfg> show config policy
{
  "rtbrick-config:policy": {
    "statement": [
      {
        "name": "IGMP_FILTER",
        "ordinal": [
          {
            "ordinal": 1,
            "description": "IGMP group filter",
            "match-logic": "or",
            "match": {
              "rule": [
                {
                  "rule": 1,
                  "type": "ipv4-mcast-group",
                  "value-type": "discrete",
                  "match-type": "or-longer",
                  "value": "198.51.100.0/24"
                },
                {
                  "rule": 2,
                  "type": "ipv4-mcast-group",
                  "value-type": "discrete",
                  "match-type": "or-longer",
                  "value": "198.51.100.0/24"
                }
              ]
            },
            "action": {
              "rule": [
                {
                  "rule": 1,
                  "operation": "return-permit"
                }
              ]
            }
          }
        ]
      }
    ]
  }
}
```

```
supervisor@leaf1: cfg> show config multicast-options igmp
{
  "rtbrick-config:igmp": {
    "interface-profile": [
      {
        "profile-name": "PROFILE1",
        "filter-policy": "IGMP_FILTER"
      }
    ]
  }
}
```

Example 4: Community Deletion Configuration

```
supervisor@rtbrick: cfg> show config policy
{
  "rtbrick-config:policy": {
    "statement": [
      {
        "name": "test_policy",
        "ordinal": [
          {
            "ordinal": 1,
            "match-logic": "or",
            "match": {
              "rule": [
                {
                  "rule": 1,
                  "type": "community",
                  "value-type": "discrete",
                  "match-type": "exists",
                  "value": [
                    "1:1"
                    ]
                },
                {
                  "rule": 2,
                  "type": "extended-community",
                  "value-type": "discrete",
                  "match-type": "exists",
                  "value": [
                    "target:3.3.3.3:3000",
                    "target:5.5.5.5:5000"
                    ]
                },
                {
                  "rule": 3,
                  "type": "large-community",
                  "value-type": "discrete",
```

```
                "match-type": "exists",
                "value": [
                  "3333:33333:33",
                  "4444:44444:44"
                  ]
              },
              {
                "rule": 4,
                "type": "community",
                "value-type": "discrete",
                "match-type": "regex",
                "value": [
                  "555:*"
                  ]
              }
            ]
          },
          "action": {
            "rule": [
              {
                "rule": 1,
                "type": "community",
                "operation": "delete-attribute-match"
              },
              {
                "rule": 2,
                "type": "extended-community",
                "operation": "delete-attribute-match"
              },
              {
                "rule": 3,
                "type": "large-community",
                "operation": "delete-attribute-match"
              },
              {
                "rule": 4,
                "operation": "return-permit"
              }
            ]
          }
        }
      ]
    }
  ]
}
}
```

**Matching ARP-ND Source in Policy**

The example below configures a policy statement named ARP_ND_SOURCE_MATCH. It defines a

rule (ordinal 1) that matches traffic of type arp-nd and appends the community 111:111 to it, with a final rule (ordinal 99) set to permit all other traffic.

```
set policy statement ARP_ND_SOURCE_MATCH
set policy statement ARP_ND_SOURCE_MATCH ordinal 1
set policy statement ARP_ND_SOURCE_MATCH ordinal 1 description
ARP_ND_POLICY_SOURCE_COM_ADD
set policy statement ARP_ND_SOURCE_MATCH ordinal 1 match
set policy statement ARP_ND_SOURCE_MATCH ordinal 1 match rule 1
set policy statement ARP_ND_SOURCE_MATCH ordinal 1 match rule 1 type source
set policy statement ARP_ND_SOURCE_MATCH ordinal 1 match rule 1 value-type discrete
set policy statement ARP_ND_SOURCE_MATCH ordinal 1 match rule 1 match-type exact
set policy statement ARP_ND_SOURCE_MATCH ordinal 1 match rule 1 value arp-nd
set policy statement ARP_ND_SOURCE_MATCH ordinal 1 action
set policy statement ARP_ND_SOURCE_MATCH ordinal 1 action rule 1
set policy statement ARP_ND_SOURCE_MATCH ordinal 1 action rule 1 type community
set policy statement ARP_ND_SOURCE_MATCH ordinal 1 action rule 1 operation append
set policy statement ARP_ND_SOURCE_MATCH ordinal 1 action rule 1 value 111:111
set policy statement ARP_ND_SOURCE_MATCH ordinal 99
set policy statement ARP_ND_SOURCE_MATCH ordinal 99 action
set policy statement ARP_ND_SOURCE_MATCH ordinal 99 action rule 99
set policy statement ARP_ND_SOURCE_MATCH ordinal 99 action rule 99 operation return-
permit
```

```
{
  "rtbrick-config:statement": [
    {
      "name": "ARP_ND_SOURCE_MATCH",
      "ordinal": [
        {
          "ordinal": 1,
          "description": "ARP_ND_POLICY_SOURCE_COM_ADD",
          "match": {
            "rule": [
              {
                "rule": 1,
                "type": "source",
                "value-type": "discrete",
                "match-type": "exact",
                "value": [
                  "arp-nd"
                  ]
              }
            ]
          },
          "action": {
            "rule": [
              {
                "rule": 1,
```

```
                "type": "community",
                "operation": "append",
                "value": [
                  "111:111"
                  ]
              }
            ]
          }
        },
        {
          "ordinal": 99,
          "action": {
            "rule": [
              {
                "rule": 99,
                "operation": "return-permit"
              }
            ]
          }
        }
      ]
    }
  ]
}
```

### 2.9.3. Policy Operational Commands

#### Policy Show Commands

#### Viewing Policy Server Statistics

The `show policy statistics policy-server` command lets you view detailed policy statistics on the policy server. The filter options let you specify a particular policy or client context to view detailed statistics, including configuration time, compile time, compilation status, and client-context subscription details.

**Syntax:**

**show policy statistics** <bd>
**show policy statistics** <bd> **app-key** <app-key>
**show policy statistics** <bd> **app-key** <app-key> **detail**
**show policy statistics** <bd> **app-key** <app-key> **policy-name** <policy-name>
**show policy statistics** <bd> **app-key** <app-key> **policy-name** <policy-name> **detail**
**show policy statistics** <bd> **detail**
**show policy statistics** <bd> **policy-name** <policy-name>
**show policy statistics** <bd> **policy-name** <policy-name> detail
**show policy statistics** policy-server
**show policy statistics policy-server detail**
**show policy statistics policy-server** <bd>

**show policy statistics policy-server** <bd> **client-context** <client-context>
**show policy statistics policy-server** <bd> **client-context** <client-context> **detail**
**show policy statistics policy-server** <bd> **client-context** <client-context> policy-name <policy-name>
**show policy statistics policy-server** <bd> **client-context** <client-context> policy-name <policy-name> **detail**
**show policy statistics policy-server** <bd> **detail**
**show policy statistics policy-server** <bd> **policy-name** <policy-name>
**show policy statistics policy-server** <bd> **policy-name** <policy-name> **client-context** <client-context>
**show policy statistics policy-server** <bd> **policy-name** <policy-name> **client-context** <client-context> **detail**
**show policy statistics policy-server** <bd> **policy-name** <policy-name> detail
**show policy statistics policy-server policy-name** <policy-name>
**show policy statistics policy-server policy-name** <policy-name> **detail**

| Option | Description |
|---|---|
| `detail` | Displays detailed information for the policy server. |
| `<bd>` | Specifies the name of the brick daemon. For example, bgp.appd.1 |
| `<app-key>` | Specifies the application-specific policy context. |
| `policy-server detail` | Displays detailed statistics for all policies on the policy server. |
| `policy-name <name>` | Displays statistics for a specific policy. |
| `policy-name <name> detail` | Displays detailed statistics for a specific policy. |
| `client-context <context>` | Displays statistics for a specific client-context. |
| `client-context <context> detail` | Displays detailed statistics for a specific client-context. |

The example below shows the statistics for all policies on the policy server.

```
supervisor@rtbrick.net: cfg>  show policy statistics policy-server
Policy Code Generation Statistics:

 Policy Name: v4_labeled_unicast_prefixs
  Configuration Time : Tue Apr 28 07:13:10 GMT +0000 2026
  Compile Start Time : Tue Apr 28 07:13:20 GMT +0000 2026
  Compile End Time   : Tue Apr 28 07:13:21 GMT +0000 2026
  Compilation Status : success

 Policy Name: v4_mvpn
  Configuration Time : Tue Apr 28 07:13:10 GMT +0000 2026
  Compile Start Time : Tue Apr 28 07:13:14 GMT +0000 2026
  Compile End Time   : Tue Apr 28 07:13:14 GMT +0000 2026
  Compilation Status : success

 Policy Name: v4_unicast_prefixs
```

```
   Configuration Time : Tue Apr 28 07:13:10 GMT +0000 2026
   Compile Start Time : Tue Apr 28 07:13:21 GMT +0000 2026
   Compile End Time   : Tue Apr 28 07:13:21 GMT +0000 2026
   Compilation Status : success

  Policy Name: v4_vrf_blue_unicast_prefixs
   Configuration Time : Tue Apr 28 07:13:10 GMT +0000 2026
   Compile Start Time : Tue Apr 28 07:13:21 GMT +0000 2026
   Compile End Time   : Tue Apr 28 07:13:22 GMT +0000 2026
   Compilation Status : success

  Policy Name: v6_labeled_unicast_prefixs
   Configuration Time : Tue Apr 28 07:13:10 GMT +0000 2026
   Compile Start Time : Tue Apr 28 07:13:22 GMT +0000 2026
   Compile End Time   : Tue Apr 28 07:13:23 GMT +0000 2026
   Compilation Status : success

  Policy Name: v6_unicast_prefixs
   Configuration Time : Tue Apr 28 07:13:10 GMT +0000 2026
   Compile Start Time : Tue Apr 28 07:13:23 GMT +0000 2026
   Compile End Time   : Tue Apr 28 07:13:24 GMT +0000 2026
   Compilation Status : success
<...>
```

The example below shows the detailed statistics for all policies on the policy server, including the path to the compiled policy library.

```
supervisor@rtbrick.net: cfg> show policy statistics policy-server detail
Policy Code Generation Statistics:

 Policy Name: v4_labeled_unicast_prefixs
  Configuration Time : Tue Apr 28 07:13:10 GMT +0000 2026
  Compile Start Time : Tue Apr 28 07:13:20 GMT +0000 2026
  Compile End Time   : Tue Apr 28 07:13:21 GMT +0000 2026
  Compilation Status : success
  Path               : /usr/share/rtbrick/libpolicy/libs/v4_labeled_unicast_prefixs.so

 Policy Name: v4_mvpn
  Configuration Time : Tue Apr 28 07:13:10 GMT +0000 2026
  Compile Start Time : Tue Apr 28 07:13:14 GMT +0000 2026
  Compile End Time   : Tue Apr 28 07:13:14 GMT +0000 2026
  Compilation Status : success
  Path               : /usr/share/rtbrick/libpolicy/libs/v4_mvpn.so

 Policy Name: v4_unicast_prefixs
  Configuration Time : Tue Apr 28 07:13:10 GMT +0000 2026
  Compile Start Time : Tue Apr 28 07:13:21 GMT +0000 2026
  Compile End Time   : Tue Apr 28 07:13:21 GMT +0000 2026
  Compilation Status : success
  Path               : /usr/share/rtbrick/libpolicy/libs/v4_unicast_prefixs.so
```

```
 Policy Name: v4_vrf_blue_unicast_prefixs
  Configuration Time : Tue Apr 28 07:13:10 GMT +0000 2026
  Compile Start Time : Tue Apr 28 07:13:21 GMT +0000 2026
  Compile End Time   : Tue Apr 28 07:13:22 GMT +0000 2026
  Compilation Status : success
  Path               :
/usr/share/rtbrick/libpolicy/libs/v4_vrf_blue_unicast_prefixs.so

<...>
```

The example below shows the detailed statistics for a specific policy daemon, including client-context subscription details.

```
supervisor@rtbrick.net: cfg> show policy statistics policy-server bgp.appd.1
Policy Code Generation Statistics:

 Policy Name: v4_labeled_unicast_prefixs
  Configuration Time : Tue Apr 28 07:13:10 GMT +0000 2026
  Compile Start Time : Tue Apr 28 07:13:20 GMT +0000 2026
  Compile End Time   : Tue Apr 28 07:13:21 GMT +0000 2026
  Compilation Status : success
  Client-contexts:
   Client-context     : export.default.PE2.ipv4.labeled-unicast
    Subscription Time : Tue Apr 28 07:13:10 GMT +0000 2026
    Notification Time : Tue Apr 28 07:13:24 GMT +0000 2026

 Policy Name: v4_unicast_prefixs
  Configuration Time : Tue Apr 28 07:13:10 GMT +0000 2026
  Compile Start Time : Tue Apr 28 07:13:21 GMT +0000 2026
  Compile End Time   : Tue Apr 28 07:13:21 GMT +0000 2026
  Compilation Status : success
  Client-contexts:
   Client-context     : export.default.PE2.ipv4.unicast
    Subscription Time : Tue Apr 28 07:13:10 GMT +0000 2026
    Notification Time : Tue Apr 28 07:13:24 GMT +0000 2026

 Policy Name: v6_labeled_unicast_prefixs
  Configuration Time : Tue Apr 28 07:13:10 GMT +0000 2026
  Compile Start Time : Tue Apr 28 07:13:22 GMT +0000 2026
  Compile End Time   : Tue Apr 28 07:13:23 GMT +0000 2026
  Compilation Status : success
  Client-contexts:
   Client-context     : export.default.PE2.ipv6.labeled-unicast
    Subscription Time : Tue Apr 28 07:13:10 GMT +0000 2026
    Notification Time : Tue Apr 28 07:13:24 GMT +0000 2026
```

The example below shows the detailed statistics for a specific policy and client-context, including subscription and notification times.

```
supervisor@rtbrick.net: cfg> show policy statistics policy-server bgp.appd.1 policy-
name v4_labeled_unicast_prefixs
Policy Code Generation Statistics:

 Policy Name: v4_labeled_unicast_prefixs
  Configuration Time : Tue Apr 28 07:13:10 GMT +0000 2026
  Compile Start Time : Tue Apr 28 07:13:20 GMT +0000 2026
  Compile End Time   : Tue Apr 28 07:13:21 GMT +0000 2026
  Compilation Status : success
  Client-contexts:
   Client-context     : export.default.PE2.ipv4.labeled-unicast
    Subscription Time : Tue Apr 28 07:13:10 GMT +0000 2026
    Notification Time : Tue Apr 28 07:13:24 GMT +0000 2026
```

### Policy Clear Commands

#### Clearing Policy Statistics

The `clear policy statistics` command removes statistics data for specified brick daemons, application keys, or policies.

**Syntax:**

**clear policy statistics** <bd>
**clear policy statistics** <bd> **app-key** <app-key>
**clear policy statistics** <bd> **app-key** <app-key> **policy-name** <policy-**name**>
**clear policy statistics** <bd> policy-name <policy-name>
**clear policy statistics policy-server**
**clear policy statistics policy-server policy-name** <policy-name>

| Option | Description |
|---|---|
| `<bd>` | Specifies the name of the brick daemon. For example, bgp.appd.1 |
| `<app-key>` | Specifies the application-specific policy context. |
| `policy-name <name>` | Specifies the name of the policy. |

The following example shows how to clear the statistics for a specific policy on the policy server:

```
supervisor@rtbrick.net: cfg> clear policy statistics policy-server policy-name
BGP_EXPORT_POLICY
```

### Policy Test

You can use the policy test feature to test a policy before attaching it to a protocol or an instance.

Perform the following tasks:

- Step 1: Identify the brick daemon that will process the policy and the table to which the policy will be applied.
- Step 2: Execute the 'test policy run' command.

Example: Testing a BGP VPN export policy

```
supervisor@leaf1: op> test policy run bgp.appd.1 policy-name VPN_V4_EXPORT table
default.bgp.rib-in.import.ipv4.vpn-unicast
```

- Step 3: View the test results.

The policy test feature will create two result tables. The result table ending with ".policy.permit" will show all objects permitted by the policy, the one ending with ".policy.deny" will show all objects denied by the policy.

Example: Viewing the result tables

```
supervisor@leaf1: op> show datastore bgp.appd.1 table default.bgp.rib-
in.import.ipv4.vpn-unicast.policy.permit
<...>
supervisor@leaf1: op> show datastore bgp.appd.1 table default.bgp.rib-
in.import.ipv4.vpn-unicast.policy.deny
<...>
```

- Step 4: Clear the result tables

You can clear the result tables using the 'test policy clear' command. Apply the clear command to the same table for which you have run the policy test.

Example: Clearing the result tables

```
supervisor@leaf1: op> test policy clear bgp.appd.1 policy-name VPN_V4_EXPORT table
default.bgp.rib-in.import.ipv4.vpn-unicast
```

# 2.10. Static Routing

## 2.10.1. Static Routing Overview

Static routing allows a network administrator to configure routes manually. Using the RtBrick CLI, you can configure static IPv4, IPv6, MPLS, and multicast routes.

For information about Route Preference, see section Route Preference.

#### Guidelines and Limitations

- ECMP is not supported for static routes that include a `null0` next-hop. If `null0` is set as one of the

next-hops, the system generates an error.

- An exit interface configuration is allowed in the case of null0, or the next hop must be link-local.

#### Supported Platforms

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

### 2.10.2. Static Routing Configuration

#### Configuration Hierarchy

The diagram illustrates the static routes configuration hierarchy.

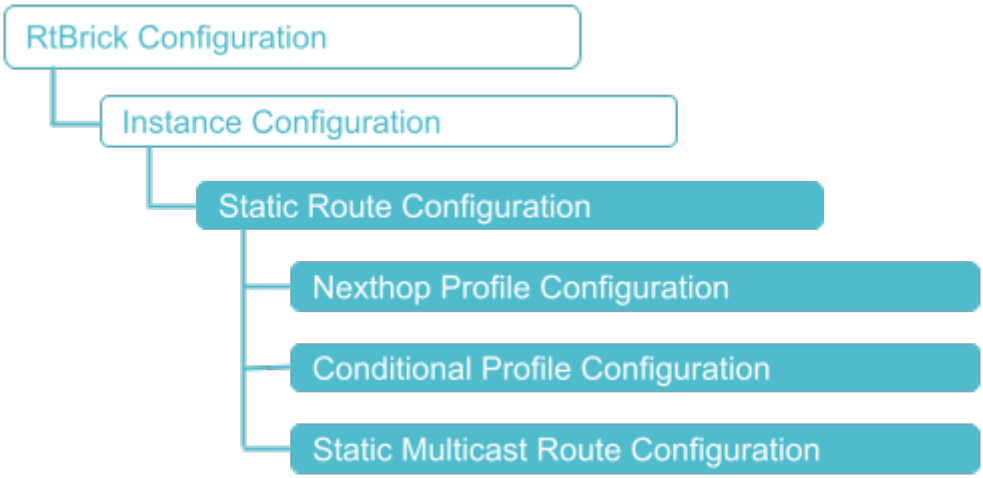



*Figure 43. Static Route Configuration Hierarchy*

#### Configuration Syntax and Commands

The following sections describe the static route configuration syntax and commands. In RBFS, next hops of static routes are configured separately, and referenced by the actual routes.

##### Static Route Configuration

This section describes how to configure the static route itself.

#### Syntax

**set instance** <instance-name> **static route** <attribute> <value>

| Attribute | Description |
|---|---|
| `<instance-name>` | Name of the routing instance |
| `<afi> <prefix \| label> (true \| false)` | Supported AFIs are ipv4, ipv6, and mpls. In case of IPv4 or IPv6, configure the prefix. In case of MPLS, configure the incoming label and BOS flag. |
| `<safi>` | Supported SAFIs are unicast, labeled-unicast, and multicast. |

| Attribute | Description |
| --- | --- |
| `<nexthop-profile>` | Name of the nexthop profile |
| `route-options no-install` | This option keeps the routes in the routing information base, but prevents them from being used for forwarding traffic. |

Example: Static Route Configuration

```
{
    "rtbrick-config:route": {
      "ipv4": [
        {
          "prefix4": "198.51.100.15/24",
          "safi": "unicast",
          "nexthop-profile": "nexthop1",
          "preference": 20

        }
      ],
      "ipv6": [
        {
          "prefix6": "2001:db8:0:117::/32",
          "safi": "unicast",
          "nexthop-profile": "nexthop2"
        }
      ],
      "mpls": [
        {
          "in-label": 8888,
          "in-bos": "true",
          "safi": "unicast",
          "nexthop-profile": "nexthop1"
        }
      ]
    }
  }
```

Example: Static Route Configuration with route-options no install

```
supervisor@rtbrick.net: cfg> show config instance default static route ipv4
192.1.0.6/32
{
  "rtbrick-config:ipv4": [
    {
      "prefix4": "192.1.0.6/32",
      "safi": "labeled-unicast",
      "nexthop-profile": "nh1",
      "route-options": "no-install"
    },
```

```
    {
      "prefix4": "192.1.0.6/32",
      "safi": "unicast",
      "nexthop-profile": "nh1"
    }
  ]
}
```

To access the RESTCONF API that corresponds to this CLI, click here.

#### Nexthop Profile Configuration

You can group various nexthop parameters with a nexthop profile name and instance, and associate this nexthop profile with multiple routes.

> A static route conditional profile check includes the route and its associated condition.

### Syntax

**set instance** <instance-name> **static nexthop-profile** <name> <attribute> <value>

| Attribute | Description |
|---|---|
| `<instance-name>` | Name of the routing instance |
| `nexthop-profile <name>` | Nexthop profile name |
| `exit-interface <exit-interface>` | Exit interface name |
| `lookup-afi (ipv4 \| ipv6 \| mpls)` | Lookup routing table address family where the nexthop will be resolved. |
| `lookup-instance <lookup-instance>` | Lookup routing table instance where the nexthop will be resolved. |
| `lookup-safi (labeled-unicast \| multicast \| unicast)` | Lookup routing table subsequent address family where the nexthop will be resolved. |
| `nexthop <address>` | IPv4/IPv6 nexthop address |
| `out-bos (true \| false)` | Label BOS |
| `out-label <out-label>` | Label to be pushed |
| `resolve-direct (true \| false)` | The option restricts all routes from resolving the nexthop of a static route instead, it allows only the direct routes to resolve the nexthop of a static route. The static nexthop profile is enabled by default. To allow indirect resolution, configure `false`. |

Example: Nexthop Profile Configuration

```
{
    "rtbrick-config:static": {
      "nexthop-profile": [
        {
          "name": "nexthop1",
          "nexthop": "198.51.100.145",
          "out-label": 4444
        },
        {
          "name": "nexthop3",
          "exit-interface": "ifp-0/0/4/4"
        }
      ]
    }
  }
```

- If you do not provide lookup-instance, lookup-afi and lookup-safi values, default values will be used to install the route.
- The exit interface attribute is mandatory for link-local nexthop.

API To access the RESTCONF API that corresponds to this CLI (IPv4 Nexthop Profile), click here.

API To access the RESTCONF API that corresponds to this CLI (IPv6 Nexthop Profile), click here.

#### Static Multicast Route Configuration

**Syntax:**

**set instance** <instance-name> **static route multicast4** <attribute> <value

| Attribute | Description |
|---|---|
| <instance-name> | Name of the routing instance |
| <source> | IPv4 multicast source address |
| <group> | IPv4 multicast group address |

Example: Static Multicast Route Configuration

```
{
    "rtbrick-config:static": {
      "route": {
        "multicast4": [
          {
            "source": "198.51.100.15/24",
            "group": "198.51.100.35/24",
            "nexthop-profile": "nexthop3"
          }
        ]
```

```
        }
      }
  }
```

To access the RESTCONF API that corresponds to this CLI, click here.

#### Conditional Profile Configuration

By using the conditional static route feature, you can make specific routes conditional. These conditional routes are installed only if the specified condition is satisfied.

You can group various conditional parameters such as compare-operation, compare-type, compare-value, match-afi, match-instance, match-interface, match-isis-level, match-peer, and match-safi, and associate this conditional profile with multiple routes.

> A static route conditional profile check includes the compare type and its associated attributes.

This feature is supported on the following Q2A and Q2C platforms:

- Edgecore CSR440 (AS7535-28XB)
- Edgecore AGR400 (AS7946-30XB)
- Edgecore AGR420 (AS7946-74XKSB)
- UfiSpace S9510-28DC
- UfiSpace S9600-32X
- UfiSpace S9600-72XC
- UfiSpace S9600-102XC

#### Configuring Conditional Profiles

**Syntax:**

**set instance** <instance-name> **static conditional-profile** <name> <attribute> <value>

| Attribute | Description |
|---|---|
| conditional-profile <name> | Name of the conditional profile |
| `compare-operation greater-than` | Conditional routing compare operation |

| Attribute | Description |
|---|---|
| compare-type route-count | Conditional routing compare type<br><br>The following compare types are supported:<br><br>• any-isis-neighbor-up<br>• bgp-peer-established<br>• interface-up<br>• isis-neighbor-up<br>• route-count |
| `compare-value <compare-value>` | Conditional routing condition value |
| match-afi (ipv4 \|ipv6 \|mpls) | Routing tables address family (AFI) for which the condition will be checked. |
| match-instance <instance-name> | Routing instance where the condition will be checked. |
| match-interface <interface-name> | Interface name for which the condition will be checked. |
| match-isis-level <level-1 \| level-2> | IS-IS level for which the condition will be checked. |
| match-peer <ipv4 \| ipv6> | BGP peer for which the condition will be checked. |
| match-peer <ipv4 \| ipv6> peer-address | BGP peer address for which the condition will be checked. |
| match-peer <ipv4 \| ipv6> source-address | BGP source address for which the condition will be checked. |
| `match-safi (labeled-unicast \|multicast \|unicast)` | Routing table subsequent address family (SAFI) for which the condition will be checked. |
| nexthop-count | This option enables the static conditional policy logic to consider the number of next hops installed in the forwarding plane for static route install decisions. |

Example 1: The following example configuration defines a conditional profile named `backbone_v4` for the `vrf1-blue` instance, matching IPv4 unicast routes, and triggers when the route count exceeds 5. It specifies the matching criteria and comparison operation for static route conditions.

```
set instance vrf1-blue static conditional-profile backbone_v4
set instance vrf1-blue static conditional-profile backbone_v4 match-instance vrf1-blue
set instance vrf1-blue static conditional-profile backbone_v4 match-afi ipv4
set instance vrf1-blue static conditional-profile backbone_v4 match-safi unicast
set instance vrf1-blue static conditional-profile backbone_v4 compare-type route-count
set instance vrf1-blue static conditional-profile backbone_v4 compare-operation
greater-than
set instance vrf1-blue static conditional-profile backbone_v4 compare-value 5
```

```
{
  "rtbrick-config:conditional-profile": [
    {
      "name": "backbone_v4",
      "match-instance": "vrf1-blue",
      "match-afi": "ipv4",
      "match-safi": "unicast",
      "compare-type": "route-count",
      "compare-operation": "greater-than",
      "compare-value": 5
    }
  ]
}
```

Example 2: The following example configuration defines a conditional static routing profile named `vrf1-blue_fabric_isis_link1` for the `vrf1-blue` instance, specifying that the static route is applied when the IS-IS neighbor on interface `ifl-0/0/0/0` in the `vrf1-blue` instance is up. It uses the `isis-neighbor-up` condition to control route activation based on IS-IS neighbor status.

```
set instance vrf1-blue static conditional-profile vrf1-blue_fabric_isis_link1
set instance vrf1-blue static conditional-profile vrf1-blue_fabric_isis_link1 match-
instance vrf1-blue
set instance vrf1-blue static conditional-profile vrf1-blue_fabric_isis_link1 compare-
type isis-neighbor-up
set instance vrf1-blue static conditional-profile vrf1-blue_fabric_isis_link1 match-
interface ifl-0/0/0/0
```

```
{
  "rtbrick-config:conditional-profile": [
    {
      "name": "vrf1-blue_fabric_isis_link1",
      "match-instance": "vrf1-blue",
      "compare-type": "isis-neighbor-up",
      "match-interface": "ifl-0/0/0/0"
    }
  ]
}
```

Example 3: The following example configuration defines a conditional static route profile named `vrf1-blue_fabric_isis_l1` for the `vrf1-blue` routing instance, which activates when any IS-IS level-1 neighbor is up and matches the `vrf1-blue` instance.

```
set instance vrf1-blue static conditional-profile vrf1-blue_fabric_isis_l1
set instance vrf1-blue static conditional-profile vrf1-blue_fabric_isis_l1 match-
instance vrf1-blue
set instance vrf1-blue static conditional-profile vrf1-blue_fabric_isis_l1 compare-
```

```
type any-isis-neighbor-up
set instance vrf1-blue static conditional-profile vrf1-blue_fabric_isis_l1 match-isis-
level level-1
```

```
{
  "rtbrick-config:conditional-profile": [
    {
      "name": "vrf1-blue_fabric_isis_l1",
      "match-instance": "vrf1-blue",
      "compare-type": "any-isis-neighbor-up",
      "match-isis-level": "level-1"
    }
  ]
}
```

Example 4: The following example configuration defines a static conditional profile named `vrf1-blue_fabric_intf_link1` for the `vrf1-blue` instance, specifying that the profile applies when the `vrf1-blue` instance is matched and the interface `ifl-0/0/0/0` is in the "up" state. It uses compare-type `interface-up` to trigger the condition based on the interface status.

```
set instance vrf1-blue static conditional-profile vrf1-blue_fabric_intf_link1
set instance vrf1-blue static conditional-profile vrf1-blue_fabric_intf_link1 match-
instance vrf1-blue
set instance vrf1-blue static conditional-profile vrf1-blue_fabric_intf_link1 compare-
type interface-up
set instance vrf1-blue static conditional-profile vrf1-blue_fabric_intf_link1 match-
interface ifl-0/0/0/0
```

```
{
  "rtbrick-config:conditional-profile": [
    {
      "name": "vrf1-blue_fabric_intf_link1",
      "match-instance": "vrf1-blue",
      "compare-type": "interface-up",
      "match-interface": "ifl-0/0/0/0"
    }
  ]
}
```

Example 5: The following example configuration defines a static conditional profile named `default_fabric_bgp_v4_198.51.100.1_peer` within the `default` instance, specifying that it matches when a BGP peer session is established with the IPv4 peer at address `198.51.100.1` and source address `198.51.100.2`. It sets the matching instance and compare type to ensure the profile applies only under these BGP session conditions.

```
set instance default static conditional-profile
```

```
default_fabric_bgp_v4_198.51.100.1_peer
set instance default static conditional-profile
default_fabric_bgp_v4_198.51.100.1_peer match-instance default
set instance default static conditional-profile
default_fabric_bgp_v4_198.51.100.1_peer compare-type bgp-peer-established
set instance default static conditional-profile
default_fabric_bgp_v4_198.51.100.1_peer match-peer ipv4 peer-address 198.51.100.1
set instance default static conditional-profile
default_fabric_bgp_v4_198.51.100.1_peer match-peer ipv4 source-address 198.51.100.2
```

```
{
  "rtbrick-config:conditional-profile": [
    {
      "name": "default_fabric_bgp_v4_198.51.100.1_peer",
      "match-instance": "default",
      "compare-type": "bgp-peer-established",
      "match-peer": {
        "ipv4": {
          "peer-address": "198.51.100.1",
          "source-address": "198.51.100.2"
        }
      }
    }
  ]
}
```

Example 6: The following example configuration defines a static conditional profile named `default_fabric_bgp_v6_2001:db8::1_peer` for the `default` instance, specifying that it matches when a BGP peer with IPv6 address `2001:db8::1` and source address `2001:db8::2` is established. It uses the `bgp-peer-established` compare type to trigger actions based on the BGP session state.

```
set instance default static conditional-profile default_fabric_bgp_v6_2001:db8::1_peer
set instance default static conditional-profile default_fabric_bgp_v6_2001:db8::1_peer
match-instance default
set instance default static conditional-profile default_fabric_bgp_v6_2001:db8::1_peer
compare-type bgp-peer-established
set instance default static conditional-profile default_fabric_bgp_v6_2001:db8::1_peer
match-peer ipv6 peer-address 2001:db8::1
set instance default static conditional-profile default_fabric_bgp_v6_2001:db8::1_peer
match-peer ipv6 source-address 2001:db8::2
```

```
{
  "rtbrick-config:conditional-profile": [
    {
      "name": "default_fabric_bgp_v6_2001:db8::1_peer",
      "match-instance": "default",
      "compare-type": "bgp-peer-established",
```

```
      "match-peer": {
        "ipv6": {
          "peer-address": "2001:db8::1",
          "source-address": "2001:db8::2"
        }
      }
    }
  ]
}
```

Example 7: The following example configuration defines a static conditional profile named `default_fabric_bgp_v6_ll_link1_peer` for the `default` instance, specifying that it matches the default instance, uses the `bgp-peer-established` compare type, and applies to the interface `ifl-0/0/1/0`. It is likely used to control routing or BGP behavior based on the state of a specific BGP peer on a particular interface.

```
set instance default static conditional-profile default_fabric_bgp_v6_ll_link1_peer
set instance default static conditional-profile default_fabric_bgp_v6_ll_link1_peer
match-instance default
set instance default static conditional-profile default_fabric_bgp_v6_ll_link1_peer
compare-type bgp-peer-established
set instance default static conditional-profile default_fabric_bgp_v6_ll_link1_peer
match-interface ifl-0/0/1/0
```

```
{
  "rtbrick-config:conditional-profile": [
    {
      "name": "default_fabric_bgp_v6_ll_link1_peer",
      "match-instance": "default",
      "compare-type": "bgp-peer-established",
      "match-interface": "ifl-0/0/1/0"
    }
  ]
}
```

Example 8: The following example configuration defines a conditional profile named 'nexthop_count_ipv6'. It sets matching criteria based on address family (IPv6) and subsequent address family (unicast), and specifies that the profile triggers when the next-hop adjacency count is greater than 53.

```
set instance default static conditional-profile nexthop_count_ipv6
set instance default static conditional-profile nexthop_count_ipv6 match-afi ipv6
set instance default static conditional-profile nexthop_count_ipv6 match-safi unicast
set instance default static conditional-profile nexthop_count_ipv6 compare-type
nexthop-count
set instance default static conditional-profile nexthop_count_ipv6 compare-operation
greater-than
```

```
set instance default static conditional-profile nexthop_count_ipv6 compare-value 53
```

```
{
  "rtbrick-config:conditional-profile": [
    {
      "name": "nexthop_count_ipv6",
      "match-afi": "ipv6",
      "match-safi": "unicast",
      "compare-type": "nexthop-count",
      "compare-operation": "greater-than",
      "compare-value": 53
    }
  ]
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

## 2.10.3. Static Routing Operational Commands

### Show Commands

#### Static Routes Created by staticd

These commands show static routes as created by the static route daemon (staticd).

**Syntax:**

**show static route** <options>

| Attribute | Description |
|---|---|
| `<afi>` | Supported AFIs are ipv4, ipv6, and mpls. |
| `<safi>` | Supported SAFIs are unicast, labeled-unicast, and multicast. |
| `instance <name>` | Static routes for an instance |

Example: List static routes for all instances

```
supervisor@rtbrick: op> show static route
Instance: default, AFI: ipv4, SAFI: unicast
Prefix/Label                                Pref    Next Hop                                    Interface
198.51.100.11/32                            2       192.0.2.2                                   ifl-0/1/20/0
198.51.100.12/32                            2       192.0.2.2                                   ifl-0/1/20/0
198.51.100.13/32                            2       192.0.2.2                                   ifl-0/1/20/0
Instance: default, AFI: mpls, SAFI: unicast
Prefix/Label                                Pref    Next Hop                                    Interface
20001,bos:1                                 2       192.0.2.2                                   ifl-0/1/20/0
20003,bos:1                                 2       192.0.2.2                                   ifl-0/1/20/0
```

```
20005,bos:1                                2       192.0.2.2                                ifl-0/1/20/0
```

#### Static Routes in the Routing Table

These commands show the static routes included in the final routing table.

**Syntax:**

> **show route** <options> **source static** <options>

| Attribute | Description |
|---|---|
| `<afi>` | Supported AFIs are ipv4, ipv6, and mpls. |
| `<safi>` | Supported SAFIs are unicast, labeled-unicast, and multicast. |
| `detail` | Detailed route information |
| `instance <name>` | Routing table information for a specific instance |
| `label <value>` | Destination label |
| `mpls` | Address family |
| `prefix <value>` | Destination prefix |
| `source` | Source of the routing information |

Example 1: List static routes information

```
supervisor@rtbrick: op> show route ipv4 source static
Flags: & - Imported
Instance: default, AFI: ipv4, SAFI: unicast
Flags  Prefix/Label                                Source              Pref  Next Hop
Outgoing Interface
       198.51.100.11/32                            static                 2  192.0.2.2
ifl-0/1/20/0
       198.51.100.12/32                            static                 2  192.0.2.2
ifl-0/1/20/0
       198.51.100.13/32                            static                 2  192.0.2.2
ifl-0/1/20/0
```

Example 2: List detailed static routes information

```
supervisor@rtbrick: op> show route ipv4 source static detail
Instance: default, AFI: ipv4, SAFI: unicast
  Prefix: 198.51.100.11/32
    Source: static, Preference: 2
      Next Hop: 192.0.2.2
        Covering prefix: 192.0.2.2/32
        Next Hop type: direct, Next Hop action: None
        Resolved in: default-ipv4-unicast(Primary)
        Outgoing Interface: ifl-0/1/20/0, NextHop MAC: 5c:07:58:79:0a:17
  Prefix: 198.51.100.12/32
```

```
    Source: static, Preference: 2
      Next Hop: 192.0.2.2
        Covering prefix: 192.0.2.2/32
        Next Hop type: direct, Next Hop action: None
        Resolved in: default-ipv4-unicast(Primary)
        Outgoing Interface: ifl-0/1/20/0, NextHop MAC: 5c:07:58:79:0a:17
  Prefix: 198.51.100.13/32
    Source: static, Preference: 2
      Next Hop: 192.0.2.2
        Covering prefix: 192.0.2.2/32
        Next Hop type: direct, Next Hop action: None
        Resolved in: default-ipv4-unicast(Primary)
        Outgoing Interface: ifl-0/1/20/0, NextHop MAC: 5c:07:58:79:0a:17
```

Example 3: List MPLS route information

```
supervisor@rtbrick: op> show route mpls
Flags: & - Imported
Instance: default, AFI: mpls, SAFI: unicast
Flags  Prefix/Label                                Source              Pref  Next Hop
Outgoing Interface
       20001,bos:1                                 static                 2  192.0.2.2
ifl-0/1/20/0
       20003,bos:1                                 static                 2  192.0.2.2
ifl-0/1/20/0
       20005,bos:1                                 static                 2  192.0.2.2
ifl-0/1/20/0
```

# 3. Layer 2 Services

The Layer 2 Services section of the RtBrick FullStack (RBFS) User Guides details the essential technical documentation and operational commands needed to configure and manage Layer 2 services.

Key Features:

- Ethernet Virtual Private Network (EVPN) is a Layer 2 technology akin to BGP/MPLS IP VPN, utilizing extended BGP for reachability and advertisements across Layer 2 networks. The EVPN Virtual Private Wire Service (VPWS) is a point-to-point (P2P) service based on this architecture.
- BGP-signaled L2VPN refers to classic Layer 2 Virtual Private Networks that utilize BGP signaling.
- LDP-signaled L2VPN involves traditional pseudowires that are LDP-signaled for point-to-point connections.
- L2X allows for direct local Layer 2 cross-connects to bridge ports or VLANs.

The following guides available under the Layer 2 Services section:

- EVPN-VPWS
- BGP-signaled L2VPN
- LDP-signaled L2VPN
- L2X

## 3.1. EVPN-VPWS

### 3.1.1. EVPN-VPWS Overview

Ethernet Virtual Private Network (EVPN) is a Layer 2 internetworking technology similar to BGP/MPLS IP VPN. EVPN uses extended BGP reachability information and advertisements between different Layer 2 networks at various sites in the control plane.

The EVPN Virtual Private Wire Service (VPWS) is a point-to-point (P2P) service that is built on the EVPN service architecture. EVPN-VPWS uses MPLS tunnels to traverse the backbone network. It offers a Layer 2 packet forwarding mode that connects access circuits (ACs) as per the specifications of RFC 8214.

RBFS also supports EVPN-VPWS services over LAG interfaces .

**Supported Standards**

| RFC Number | Description |
|---|---|
| 7432 | BGP MPLS-Based Ethernet VPN |
| 8214 | Virtual Private Wire Service Support in Ethernet VPN |

### EVPN VPWS Network Model

As shown in the figure below, an EVPN VPWS network contains the following building blocks:

- Customer edge (CE)—Customer device directly connected to the service provider network.
- Provider edge (PE)—Service provider device connected to CEs. PEs provide access to the EVPN VPWS network and forward traffic between customer network sites by using public tunnels.
- Attachment circuit (AC)—A physical or virtual link between a CE and a PE.
- Pseudowire (PW)—A virtual bidirectional connection between two PEs. A PW comprises a pair of virtual links in opposite directions.
- MPLS transport tunnel—A connection that carries one or more PWs across the MPLS core or IP backbone, such as an MPLS tunnel.
- Cross-connect—A connection formed by two physical or virtual circuits, such as ACs and PWs, that switches packets between them.

The figure below shows the protocol packet exchange process in the EVPN VPWS.

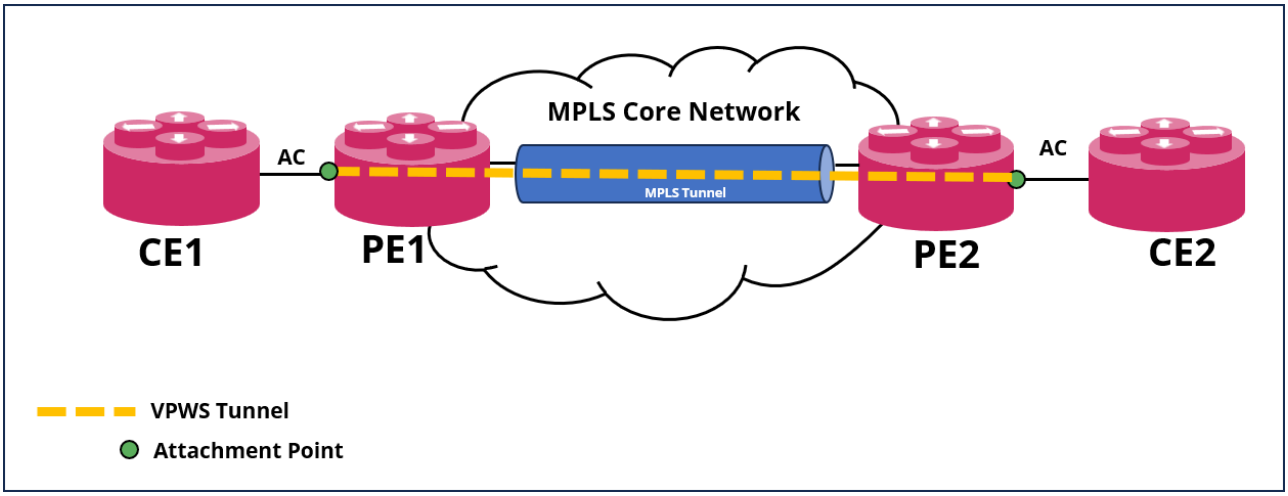



*Figure 44. Protocol Packet Exchange Process in EVPN VPWS*

PE1 and PE2 are each configured with an EVPN VPWS instance. After the PE receives packets from the AC, it adds the PW label and sends them to the peer PE through the MPLS label-switched path (LSP). After the other PE (PE2) receives the packet via the MPLS LSP, it pops the PW label of the packets and forwards the packets to the AC bound to the PW.

### VLAN Tag Manipulation

VLAN Tag Manipulation involve modifying VLAN tags on Ethernet frames to ensure proper traffic segmentation and delivery across a Layer 2 network. RBFS supports some VLAN Tag Manipulation operations which are used to manipulate tags on Ethernet frames as they enter or leave a network interface. These manipulations include adding, removing, and rewriting VLAN tags. VLAN manipulation operations can be performed both at ingress and egress nodes.

The following VLAN manipulation operations are supported for Layer 2 VPN interfaces:

- **push**: Adds a single outer VLAN tag to an Ethernet frame.
- **pop**: Removes a single outer VLAN tag from an Ethernet tag frame.

- **swap**: Rewrite a single outer VLAN tag of an Ethernet tag frame.
- **push–push**: Adds two VLAN tags to an Ethernet frame.
- **pop-pop**: Removes two VLAN tags from an Ethernet tags frame.
- **swap–swap**: Rewrite both the outer and inner VLAN tags of an Ethernet tags frame.
- **swap–push**: Rewrite the outer VLAN tag and add a new inner VLAN tag for Ethernet tags' frame.
- **pop–swap**: Removes the outer VLAN tag and rewrites the inner VLAN tag of an Ethernet tags' frame.

**Supported CLI Options for VLAN Tag Manipulation Operations**

The following table presents the various VLAN tag manipulation operations and supported CLI options.

| CLI Options | pop | push | swap | push-push | pop-pop | swap-push | swap-swap | pop-swap |
|---|---|---|---|---|---|---|---|---|
| vlan-id | No | Yes | Yes | Yes | No | Yes | Yes | Yes |
| vlan-encapsulation | No | Yes | Yes | Yes | No | Yes | Yes | Yes |
| innervlan-id | No | No | No | Yes | No | Yes | Yes | No |

**Limitations**

Currently, the following features are not supported for EVPN VPWS:

- Policy filtering options (at Address Family/Peer-Group/Instance levels)
- Route Reflection
- Add-Path
- VLAN change configuration on L2 IFL
- QoS is not supported for L2VPN-VPWS services when using LAG interfaces.

**Supported Platforms**

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

## 3.1.2. EVPN VPWS Configuration

**Configuration Hierarchy**

The diagram illustrates the EVPN-VPWS configuration hierarchy. All EVPN VPWS configurations are done within an instance, such as the default instance or an EVPN service instance.

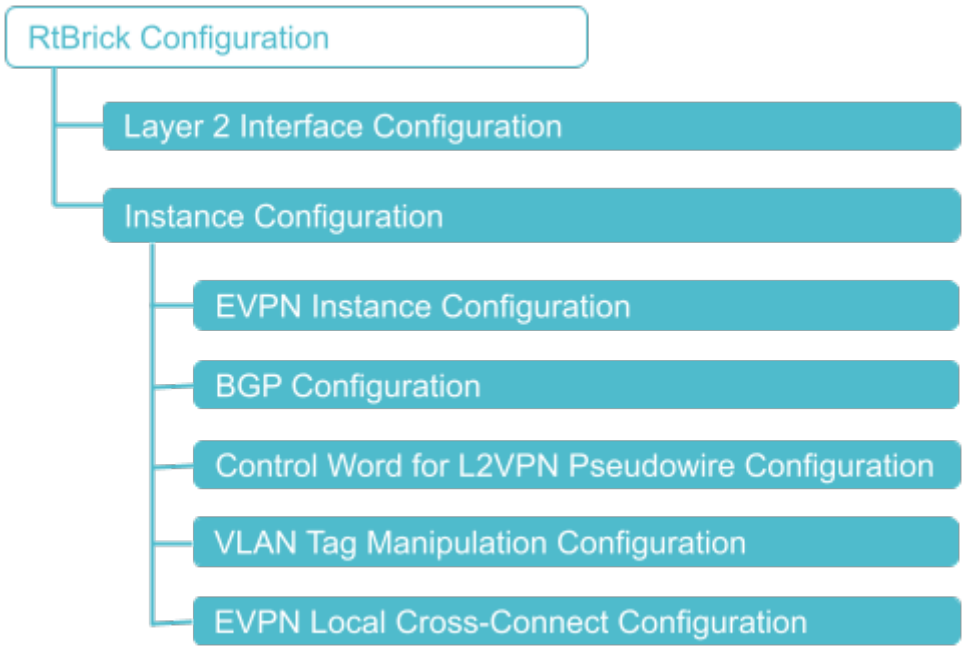



*Figure 45. EVPN VPWS Configuration Hierarchy*

### Configuration Syntax and Commands

The following sections describe the EVPN-VPWS configuration syntax and commands.

#### Layer 2 Interface Configuration

EVPN-VPWS supports the configuration of Layer 2 logical interfaces.

**Syntax:**

```
set interface <name> <attribute> <value>
```

| Attribute | Description |
|---|---|
| `<name>` | Specify the name of the interface. Example: ifp-0/0/1. |
| `unit <unit-id>` | Create a logical interface (also referred to as a sub-interface) under the physical interface |
| `unit <unit-id> interface-type l2vpn-vpws` | Specify the type of the L2VPN interface. |
| `unit <unit-id> match-type port` | Sets the "port" match-type for the L2VPN VPWS untagged interfaces. For tagged interfaces, "port" match-type is not supported. Ensure that there are no other tagged configurations for the interfaces when the port match-type is configured. When port matching is configured, it will match all traffic regardless of whether it is untagged, single-tagged, or double-tagged VLANs. |
| `unit <unit-id> vlan <outer-vlan-id>` | Outer VLAN ID. |

| Attribute | Description |
| --- | --- |
| `unit <unit-id> vlan <inner-vlan-id>` | Inner VLAN ID. |
| `unit <unit-id> instance <instance>` | Assign the logical interface to an instance. |

The following example shows an untagged interface configuration along with "port" match-type for the L2VPN VPWS untagged interface.

```
set interface ifp-0/0/17 unit 0
set interface ifp-0/0/17 unit 0 interface-type l2vpn-vpws
set interface ifp-0/0/17 unit 0 match-type port
set interface ifp-0/0/17 unit 0 instance evpn-vpws-vrf1
commit
```

The following example shows a single-tagged interface configuration.

```
set interface ifp-0/0/17 unit 100
set interface ifp-0/0/17 unit 100 interface-type l2vpn-vpws
set interface ifp-0/0/17 unit 100 instance evpn-vpws-vrf2
set interface ifp-0/0/17 unit 100 vlan 100
commit
```

The following example shows a double-tagged interface configuration.

```
set interface ifp-0/0/17 unit 200
set interface ifp-0/0/17 unit 200 interface-type l2vpn-vpws
set interface ifp-0/0/17 unit 200 instance evpn-vpws-vrf3
set interface ifp-0/0/17 unit 200 vlan 200
set interface ifp-0/0/17 unit 200 inner-vlan 201
commit
```

#### EVPN Instance Configuration

An EVPN Instance (EVI) is a routing and forwarding instance of EVPN that covers all the participating PE routers in a VPN. EVI is configured per customer on the PE routers. Each EVI has a unique route distinguisher and one or more route targets.

**Syntax:**

**set instance** <name> <attribute> <value>

| Attribute | Description |
| --- | --- |
| `<name>` | A unique name for the EVPN routing instance. |

| Attribute | Description |
|---|---|
| `address-family <afi>` | Address family identifier (AFI). Supported value: l2vpn |
| `address-family <afi> <safi>` | Subsequent address family identifier (SAFI). Supported values: evpn-vpws<br>`evpn-vpws` needs to be enabled on the VRF instance. |
| `protocol` | Specifies the routing protocol |
| `route-distinguisher as-number \| ipv4-address:id>` | The route distinguisher (RD) uniquely defines routes within an IPv4 network. PE routers use route distinguishers to identify which VPN a packet belongs to. Supported formats are <as-number:id> or <ipv4-address:id>. |
| `ipv4-router-id <ipv4-router-id>` | The router ID of the routing instance. |
| `route-target ( import \| export ) <rt-value>` | Route targets (RT) are used to transfer routes between VPN instances. The RT identifies a subset of routes that should be imported to or exported from a particular VPN instance. You can configure an RT for importing or exporting routes or both. |

In the following configuration, VRF instance AFI has been set to AFI: l2vpn and SAFI: evpn-vpws.

```
set instance evpn-vpws-vrf1
set instance evpn-vpws-vrf1 ipv4-router-id 192.1.6.3
set instance evpn-vpws-vrf1 route-distinguisher 192.1.6.3:65006
set instance evpn-vpws-vrf1 address-family l2vpn evpn-vpws
set instance evpn-vpws-vrf1 address-family l2vpn evpn-vpws route-target import
target:192.1.6.0:65006
set instance evpn-vpws-vrf1 address-family l2vpn evpn-vpws route-target export
target:192.1.6.0:65006
commit
```

#### BGP Configuration

#### BGP L2VPN VFT (Virtual Forwarding Table) Configuration

### Syntax:

**set instance** <name> **protocol bgp** <attribute> <value>

| Attribute | Description |
|---|---|
| `<name>` | Name of the routing instance |
| `address-family <afi>` | Address family identifier (AFI). Supported value: l2vpn |

| Attribute | Description |
|---|---|
| `address-family <afi> <safi>` | Subsequent address family identifier (SAFI). Supported value: evpn-vpws<br>`evpn-vpws` needs to be enabled on the VRF instance. |
| `local-as <as-number>` | The AS number in four-byte format. The numbers allowed are from 1 to 4294967295. |
| `interface <name>` | Interface that is bound to L2VPN |
| `interface <name> local-service-id <local-service-id>` | Specify the local service ID that is used to establish an EVPN PW between two PEs |
| `interface <name> remote-service-id <remote-service-id>` | Specify the remote service ID that is used to establish an EVPN PW between two PEs |

The following example configures `evpn-vpws-vrf1` instance for BGP L2VPN.

```
set instance evpn-vpws-vrf1 protocol bgp local-as 65006
set instance evpn-vpws-vrf1 protocol bgp address-family l2vpn evpn-vpws
set instance evpn-vpws-vrf1 protocol bgp address-family l2vpn evpn-vpws interface ifl-
0/0/17/0
set instance evpn-vpws-vrf1 protocol bgp address-family l2vpn evpn-vpws interface ifl-
0/0/17/0 local-service-id 100
set instance evpn-vpws-vrf1 protocol bgp address-family l2vpn evpn-vpws interface ifl-
0/0/17/0 remote-service-id 101
commit
```

#### BGP L2VPN EVPN Configuration

##### Configuring the BGP L2VPN EVPN Address Family

**Syntax:**

**set instance** <name> **protocol bgp address-family l2vpn evpn**

| Attribute | Description |
|---|---|
| `<name>` | Name of the routing instance |

To configure BGP L2VPN EVPN on the default instance, enter the following command:

```
set instance default protocol bgp address-family l2vpn evpn
commit
```

#### Configuring Address Families for Peer Groups

**Syntax:**

> **set instance** <instance-name> **protocol bgp peer-group** <pg-name> **address-family** <afi> <safi> <attribute> <value>

| Attribute | Description |
|---|---|
| `<afi>` | Address family identifier (AFI). Supported value: l2vpn. |
| `<safi>` | Subsequent address family identifier (SAFI). Supported value: evpn |
| `extended-nexthop` | Enable extended-next-hop encoding for BGP peer groups to allow the transfer of IPv4 prefixes over an IPv6 connection. |
| `update-nexthop ( ipv4-address \| ipv6-address ) <address>` | BGP nexthop address for routes advertised to this peer group |

To configure BGP L2VPN Peer Group on the `default` instance, enter the following command:

```
set instance default protocol bgp peer-group spine address-family l2vpn evpn
commit
```

#### Configuring Control Word for L2VPN Pseudowire

In a network with EVPN-VPWS service, a control word can be inserted between the label stack and the MPLS payload. Control word is disabled by default.

**Syntax**:

> **set instance** <name> **protocol bgp address-family l2vpn evpn-vpws interface** <interface> **control-word enable**

| Attribute | Description |
|---|---|
| `<name>` | Name of the routing instance. |
| `<interface>` | Name of the interface. |

To enable Control Word on the `default` instance, enter the following command:

```
set instance default protocol bgp peer-group spine address-family l2vpn evpn-vpws
interface ifl-0/0/1/10 control-word enable
commit
```

#### Configuring VLAN Tag Manipulation

The following section describes the VLAN tag manipulation configuration syntax and commands.

**Syntax:**

**set interface** <interface-name> **unit** <unit id> **[input-vlan-map | output-vlan-map]** <attribute> <value>

| Attribute | Description |
|---|---|
| `<name>` | Specify the name of the interface. |
| `unit <unit-id>` | Specify the unit ID for the interface. |
| `unit <unit-id> input-vlan-map` | Ingress VLAN mapping. |
| `unit <unit-id> output-vlan-map` | Egress VLAN mapping |
| `vlan-operation` | Specify any of the VLAN operations supported: `pop`, `pop-pop`, `pop-swap`, `push`, `push-push`, `swap`, `swap-push`, and `swap-swap`. |
| `vlan-encapsulation` | Specify the encapsulation type for the VLAN operation. Supported only 802.1q. |
| `vlan-id` | Specify the outer VLAN for the VLAN operation specified. |
| `inner-vlan-id` | Specify the inner VLAN for the VLAN operation specified. |

Example for VLAN 'push' operation configuration

```
supervisor@rtbrick.net: cfg> show config set interface ifp-0/1/4 unit 1000
set interface ifp-0/1/4 unit 1000
set interface ifp-0/1/4 unit 1000 interface-type l2vpn-vpws
set interface ifp-0/1/4 unit 1000 output-vlan-map vlan-operation push
set interface ifp-0/1/4 unit 1000 output-vlan-map vlan-id 1001
set interface ifp-0/1/4 unit 1000 instance vrf2-evpn-vpws
set interface ifp-0/1/4 unit 1000 vlan 1000
```

Example: VLAN 'push' operation configuration.

```
supervisor@rtbrick.net: cfg> show config interface ifp-0/1/4 unit 1000
{
  "rtbrick-config:unit": [
    {
      "unit-id": 1000,
      "interface-type": "l2vpn-vpws",
      "output-vlan-map": {
        "vlan-operation": "push",
        "vlan-id": 1001
      },
      "instance": "vrf2-evpn-vpws",
```

```
      "vlan": 1000
    }
  ]
}
```

#### Configuring EVPN Local Cross-Connect

Local cross-connect is a special case of VPWS in which local and remote sites are connected to the same router. When the VPWS configuration is received, BGP checks if the exact remote site is locally configured. If so, BGP understands it needs to perform a local cross-connect. Local cross-connect for the local site and remote site and vice versa should be configured on the same VRF instance.

The following example shows a single-tagged interface configuration for the VRF instance (`evpn-vpws-vrf1`).

```
set interface ifp-0/0/17 unit 0
set interface ifp-0/0/17 unit 0 interface-type l2vpn-vpws
set interface ifp-0/0/17 unit 0 instance evpn-vpws-vrf1
set interface ifp-0/0/17 unit 0 vlan 100
set interface ifp-0/0/17 unit 10
set interface ifp-0/0/17 unit 10 interface-type l2vpn-vpws
set interface ifp-0/0/17 unit 10 instance evpn-vpws-vrf1
set interface ifp-0/0/17 unit 10 vlan 100
commit
```

The following examples show the local cross-connect configurations for the same VRF instance (`evpn-vpws-vrf1`).

```
set instance evpn-vpws-vrf1 protocol bgp local-as 65006
set instance evpn-vpws-vrf1 protocol bgp address-family l2vpn evpn-vpws
set instance evpn-vpws-vrf1 protocol bgp address-family l2vpn evpn-vpws interface ifl-
0/0/17/0
set instance evpn-vpws-vrf1 protocol bgp address-family l2vpn evpn-vpws interface ifl-
0/0/17/0 local-service-id 100
set instance evpn-vpws-vrf1 protocol bgp address-family l2vpn evpn-vpws interface ifl-
0/0/17/0 remote-service-id 101
commit
```

```
set instance evpn-vpws-vrf1 protocol bgp local-as 65006
set instance evpn-vpws-vrf1 protocol bgp address-family l2vpn evpn-vpws
set instance evpn-vpws-vrf1 protocol bgp address-family l2vpn evpn-vpws interface ifl-
0/0/17/10
set instance evpn-vpws-vrf1 protocol bgp address-family l2vpn evpn-vpws interface ifl-
0/0/17/10 local-service-id 101
set instance evpn-vpws-vrf1 protocol bgp address-family l2vpn evpn-vpws interface ifl-
0/0/17/10 remote-service-id 100
```

```
commit
```

#### Enabling LAG Interfaces for EVPN VPWS Services

These following example configurations show how to collectively set up a Layer 2 VPN service using EVPN-VPWS over a link aggregation (LAG) interface.

1. Create a LAG interface
2. Create a service instance with VLAN-based attachment
3. Configure BGP-signaled EVPN VPWS instance to establish point-to-point Layer 2 connectivity between remote endpoints.

The following example configures a LAG named 'lag-1' in static mode with a minimum active link requirement of 1. It adds ifp-0/0/3 as a member interface of the LAG.

```
set link-aggregation interface lag-1
set link-aggregation interface lag-1 description static_lag
set link-aggregation interface lag-1 mode static
set link-aggregation interface lag-1 minimum-link-count 1
set link-aggregation interface lag-1 member-interface ifp-0/0/3
```

```
{
  "rtbrick-config:interface": [
    {
      "interface-name": "lag-1",
      "description": "static_lag",
      "mode": "static",
      "minimum-link-count": 1,
      "member-interface": [
        {
          "member-interface-name": "ifp-0/0/3"
        }
      ]
    }
  ]
}
```

The following example configures a logical unit 1000 on lag-1 for L2VPN VPWS service, associating it with the 'vrf1-evpn-vpws' instance. It also assigns VLAN ID 1000 to the unit for service delimitation within the VPWS instance.

```
set interface lag-1 unit 1000
set interface lag-1 unit 1000 interface-type l2vpn-vpws
set interface lag-1 unit 1000 instance vrf1-evpn-vpws
set interface lag-1 unit 1000 vlan 1000
```

```
{
  "rtbrick-config:unit": [
    {
      "unit-id": 1000,
      "interface-type": "l2vpn-vpws",
      "instance": "vrf2-evpn-vpws",
      "vlan": 1000
    }
  ]
}
```

The following example configures an EVPN VPWS instance 'vrf1-evpn-vpws' with IPv4 router ID, route distinguisher, and import/export route-targets to control the distribution of EVPN routes. It enables BGP for EVPN-VPWS signaling and binds the service to lag-1/1000. It defines local and remote service IDs to establish a point-to-point Layer 2 VPN connection between peers.

```
set instance vrf1-evpn-vpws
set instance vrf1-evpn-vpws ipv4-router-id 192.0.2.0
set instance vrf1-evpn-vpws route-distinguisher 192.0.2.0:65016
set instance vrf1-evpn-vpws address-family l2vpn evpn-vpws
set instance vrf1-evpn-vpws address-family l2vpn evpn-vpws route-target import
target:192.0.2.0:65016
set instance vrf1-evpn-vpws address-family l2vpn evpn-vpws route-target export
target:192.0.2.0:65016
set instance vrf1-evpn-vpws protocol bgp local-as 65016
set instance vrf1-evpn-vpws protocol bgp router-id 192.0.2.0
set instance vrf1-evpn-vpws protocol bgp address-family l2vpn evpn-vpws
set instance vrf1-evpn-vpws protocol bgp address-family l2vpn evpn-vpws interface lag-
1/1000
set instance vrf1-evpn-vpws protocol bgp address-family l2vpn evpn-vpws interface lag-
1/1000 local-service-id 1000
set instance vrf1-evpn-vpws protocol bgp address-family l2vpn evpn-vpws interface lag-
1/1000 remote-service-id 1001
```

```
{
  "rtbrick-config:instance": [
    {
      "name": "vrf1-evpn-vpws",
      "ipv4-router-id": "192.0.2.0",
      "route-distinguisher": "192.0.2.0:65016",
      "address-family": [
        {
          "afi": "l2vpn",
          "safi": "evpn-vpws",
          "route-target": {
            "import": [
              "target:192.0.2.0:65016"
              ],
```

```
            "export": [
              "target:192.0.2.0:65016"
              ]
          }
        }
      ],
      "protocol": {
        "bgp": {
          "local-as": 65016,
          "router-id": "192.0.2.0",
          "address-family": [
            {
              "afi": "l2vpn",
              "safi": "evpn-vpws",
              "interface": [
                {
                  "name": "lag-1/1000",
                  "local-service-id": 1000,
                  "remote-service-id": 1001
                }
              ]
            }
          ]
        }
      }
    }
  ]
}
```

### 3.1.3. EVPN VPWS Operational Commands

#### EVPN VPWS Show Commands

The following show commands display the EVPN VPWS-related information.

##### BGP Summary

This command displays BGP protocol parameters like attributes or timers that are generic to the BGP instance.

**Syntax:**

**show bgp summary** <option>

| Attribute | Description |
|---|---|
| Option | Description |

| Attribute | Description |
|---|---|
| - | Without any option, the command displays the information for all instances. |
| instance <instance-name> | BGP summary information for the given instance. |

Example: BGP summary for instance evpn-vpws-vrf1

```
supervisor@rtbrick: op> show bgp summary instance ce1-vrf
Instance: ce1-vrf
  General information
    Hostname: ce1-vrf instance, Domain name: ce1-vrf instance
    Local AS: 64501, Version: 4
    Local preference: 100, eBGP Protocol preference: 20, iBGP Protocol preference: 200
    Router ID: 198.51.100.201, Cluster ID: 198.51.100.201
    MED: 0, Enforce first AS: True
    Address families configured:
      ipv4: unicast
      ipv6: unicast
      l2vpn: evpn-vpws
  Capabilities
    Route refresh: True, AS4: True, Graceful restart: False, L2VPN EVPN-VPWS:True
  Best route selection
    Always compare MED: True, Ignore as path: False
    Ignore local preference: False, Ignore origin: False
    Ignore MED: False, Ignore route source: False
    Ignore router ID: False, Ignore uptime: True
    Ignore cluster length: False, Ignore peer IP: False
    Route select parameter: 0
  Timers
    Connect retry: 30s, Keepalive: 30s, Holdtime: 90s
  Statistics
    Peers configured: 0, Peers auto discovery: 0
      Peers in idle          : 0
      Peers in connect       : 0
      Peers in active        : 0
      Peers in opensent      : 0
      Peers in openconfirm   : 0
      Peers in established   : 0
```

#### BGP Peer

The 'show bgp peer' commands display information on BGP peers.

**Syntax:**

**show bgp peer** <option>

| Option | Description |
|---|---|
| - | Without any option, the commands display all BGP peers in all instances in a summary table format. |
| detail | Detailed information on all BGP peers in all instances in a list view. |
| <peer-name> | Detailed information on the peer with the given name. |
| address <peer-address> | Detailed information on the peer with the given IP address. |
| history | Displays BGP peer history information such as the peer state down reasons. |
| history <peer-address> | Displays BGP peer history information such as the peer state down reasons for a specified peer. |
| instance <instance-name> | Summary of all BGP peers in the given instance. |
| instance <instance-name> detail | Detailed information on all BGP peers in the given instance. |
| instance <instance-name> detail <peer-name> | Detailed information on the peer with the given name in the given instance. |
| instance <instance-name> detail address <peer-address> | Detailed information on the peer with the given IP address in the given instance. |
| statistics | Received and sent BGP prefixes per AFI/SAFI for all peers in all instances. |
| statistics peer <peer-name> | Received and sent BGP prefixes per AFI/SAFI for the peer with the given name. |
| statistics peer address <peer-address> | Received and sent BGP prefixes per AFI/SAFI for the peer with the given IP address. |
| statistics instance <instance-name> peer <peer-name> | Received and sent BGP prefixes per AFI/SAFI for the peer with the given name in the given instance. |
| statistics instance <instance-name> peer address <peer-address> | Received and sent BGP prefixes per AFI/SAFI for the peer with the given IP address in the given instance. |

Example: BGP Peer Information for instance `default`

```
supervisor@rtbrick: op> show bgp peer instance default
Instance: default
  Peer                                    Remote AS    State         Up/Down Time
PfxRcvd              PfxSent
  PE2                                     65542        Established   0d:00h:00m:09s
14                   24
```

Example: Detailed view of BGP Peer Information

```
supervisor@rtbrick: op> show bgp peer detail
```

```
Peer: PE2, Peer IP: 192.0.2.2, Remote AS: 65542, Local IP: 192.0.2.1, Local AS: 65541,
Any AS: False
  Type: ebgp, State: Established, Up/Down Time: Wed Jul 15 19:01:53 GMT +0000 2026
  Discovered on interface: -
  Last transition: Wed Jul 15 19:01:53 GMT +0000 2026, Flap count: 0
  Peer ID        : 198.51.100.21, Local ID: 198.51.100.11
  Instance       : default, Peer group: PE2
  6PE enabled    : False
  TTL security   : False, TTL limit: -
  Authentication:
    ID: BGP_AUTH_HMAC-SHA-256-128, Algorithm: HMAC-SHA-256-128
  Timer values:
    Peer keepalive : 30s, Local keepalive: 30s
    Peer holddown  : 90s, Local holddown : 90s
    Connect retry  : 30s
  Timers:
    Connect retry timer : 0
    Keepalive timer     : expires in 16s 106160us
    Holddown timer      : expires in 1m 19s 856233us
  NLRIs:
    Sent           : ['l2vpn-evpn', 'ipv4-vpn-unicast', 'ipv6-vpn-unicast', 'ipv4-
labeled-unicast', 'ipv6-labeled-unicast']
    Received       : ['l2vpn-evpn', 'ipv4-vpn-unicast', 'ipv6-vpn-unicast', 'ipv4-
labeled-unicast', 'ipv6-labeled-unicast']
    Negotiated     : ['l2vpn-evpn', 'ipv4-vpn-unicast', 'ipv6-vpn-unicast', 'ipv4-
labeled-unicast', 'ipv6-labeled-unicast']
  Capabilities:
    Addpath sent                  : None
    Addpath received              : None
    Addpath negotiated            : None
    Extended nexthop sent         : None
    Extended nexthop received     : None
    Extended nexthop negotiated   : None
    Multiple labels sent          : None
    Multiple labels received      : None
    Multiple labels negotiated    : None
    Capabilities:
      Feature                      Sent            Received        Negotiated
      Route refresh                True            True            True
      4 byte AS                    True            True            True
      Graceful restart             False           False           False
      Link local only              False           False           False
  Prefix Limit:
  End of RIB:
    Address family                 Sent                                Received
    IPv4 unicast                   never                               never
    IPv4 labeled-unicast           Wed Jul 15 19:01:59 GMT +0000 2026  Wed Jul 15
19:01:58 GMT +0000 2026
    IPv6 unicast                   never                               never
    IPv6 labeled-unicast           Wed Jul 15 19:01:59 GMT +0000 2026  Wed Jul 15
19:01:58 GMT +0000 2026
```

```
    IPv4 VPN-unicast              Wed Jul 15 19:01:59 GMT +0000 2026  Wed Jul 15
19:01:58 GMT +0000 2026
    IPv6 VPN-unicast              Wed Jul 15 19:01:59 GMT +0000 2026  Wed Jul 15
19:01:58 GMT +0000 2026
    IPv4 flowspec                 never                               never
    IPv6 flowspec                 never                               never
    IPv4 VPN-multicast            never                               never
    L2VPN VPLS                    never                               never
    L2VPN EVPN                    Wed Jul 15 19:01:59 GMT +0000 2026  Wed Jul 15
19:01:58 GMT +0000 2026
  Message stats:
    Session stats:
      Direction              Open          Update       Keepalive          Notify
Route refresh
      Input                     1              11               1               0
0
      Output                    1              11               1               0
0
    Total stats:
      Input                     1              11               1               0
0
      Output                    1              11               1               0
0
    Route stats:
      Address family                        Received            Sent    Prefix limit
Idle timeout
      IPv4 unicast                                 0               0               0
0
      IPv4 labeled-unicast                         4               6               0
0
      IPv6 unicast                                 0               0               0
0
      IPv6 labeled-unicast                         4               6               0
0
      IPv4 VPN-unicast                             2               4               0
0
      IPv6 VPN-unicast                             2               4               0
0
      IPv4 multicast                               0               0               0
0
      IPv4 VPN-multicast                           0               0               0
0
      L2VPN VPLS                                   0               0               0
0
      L2VPN EVPN                                   2               4               0
0
      IPv4 Flowspec                                0               0               0
0
      IPv6 Flowspec                                0               0               0
0
```

Example: BGP Peer Statistics for a specific peer

```
supervisor@rtbrick: op> show bgp peer statistics peer address 192.0.2.2
Instance: default
  Peer                            AFI     SAFI                PfxRcvd    PfxSent
  PE2                             ipv4    unicast             0          0
                                  ipv4    labeled-unicast     4          6
                                  ipv6    unicast             0          0
                                  ipv6    labeled-unicast     4          6
                                  ipv4    vpn-unicast         2          4
                                  ipv6    vpn-unicast         2          4
                                  ipv4    multicast           0          0
                                  ipv4    vpn-multicast       0          0
                                  l2vpn   vpls                0          0
                                  l2vpn   evpn                2          4
                                  ipv4    flowspec            0          0
                                  ipv6    flowspec            0          0
```

Example: BGP Peer history for a specified peer

```
supervisor@rtbrick: op> show bgp peer history peer address 192.0.2.2
Instance: default
  Peer Address                              Source Address
Type         Last Reset Reason  Update Time
  192.0.2.2                                 192.0.2.1
Config C...  Cease, Sub-Cod...  Wed Jul 15 19:01:24
  192.0.2.2                                 192.0.2.1
TCP Error    Cease, Sub-Cod...  Wed Jul 15 19:00:33
  192.0.2.2                                 192.0.2.1
Config C...  Cease, Sub-Cod...  Wed Jul 15 19:00:33
  192.0.2.2                                 192.0.2.1
Notifica...  Cease, Sub-Cod...  Wed Jul 15 18:59:40
  192.0.2.2                                 192.0.2.1
Clear BG...  Cease, Sub-Cod...  Wed Jul 15 18:59:37
```

#### BGP RIB-in

This command displays the received routes.

**Syntax:**

**show bgp rib-in** <option>

| Option | Description |
|---|---|
| - | Without any option, the command displays information on the received BGP routing table on all instances in a summary table format. |
| <afi> | BGP routing table summary for the given address family (AFI), all sub-address families and all instances. Supported AFI values are 'ipv4', 'ipv6' and l2vpn. |
| <afi> <safi> | BGP routing table summary for the given address family (AFI) and sub-address family (SAFI), and all instances. Supported SAFI values are evpn, 'labeled-unicast', 'unicast', 'vpn-multicast', 'vpn-unicast', and evpn-vpws. |
| <afi> <safi> detail | Detailed list view of the BGP routing table for the given address family (AFI) and sub-address family (SAFI), and all instances. |
| <afi> <safi> <prefix> | BGP routing table entry for the given prefix and all instances. |
| <afi> <safi> instance <instance-name> | BGP routing table summary for the given AFI, SAFI, and instance. |
| <afi> <safi> instance <instance-name> detail | Detailed list view of BGP routing table for the given AFI, SAFI, and instance. |
| <afi> <safi> instance <instance-name> <prefix> | BGP routing table entry for the given prefix and instance. |
| <afi> <safi> community <community-name> | BGP community details for the given AFI, SAFI, and instance. |
| <afi> <safi> error | BGP route with error status for the given AFI, SAFI, and instance. |
| <afi> <safi> peer <name> / peer address <ip> | Peer name or address |

Example: Summary view of the BGP rib-in where you can find received information for the EVPN VPWS instances like evpn-vpws-vrf1, evpn-vpws-vrf2, and evpn-vpws-vrf3.

```
supervisor@rtbrick: op> show bgp rib-in
Instance: ce1-vrf, AFI: ipv4, SAFI: unicast
  Hostname: Local, Peer IP: 0.0.0.0
  Source IP: 0.0.0.0, Total routes: 2
    Prefix                                      Next Hop
MED         Lpref       AS Path
    198.51.100.201/32                           -                                   0
-           -
    198.51.100.202/32                           -                                   0
-           -
Instance: default, AFI: ipv4, SAFI: labeled-unicast
  Hostname: Local, Peer IP: 0.0.0.0
  Source IP: 0.0.0.0, Total routes: 4
    Prefix                                      Next Hop
MED         Lpref       AS Path
    192.0.2.0/30                                -                                   0
```

```
-           -
    192.0.2.4/30                                -                                   0
-           -
    198.51.100.11/32                            -                                   0
-           -
    198.51.100.12/32                            -                                   0
-           -
Instance: default, AFI: ipv4, SAFI: labeled-unicast
  Hostname: PE2, Peer IP: 192.0.2.2
  Source IP: 192.0.2.1, Total routes: 4
    Prefix                                      Next Hop
MED         Lpref       AS Path
    192.0.2.0/30                                192.0.2.2                           0
-           65542
    192.0.2.4/30                                192.0.2.2                           0
-           65542
    198.51.100.21/32                            192.0.2.2                           0
-           65542
    198.51.100.22/32                            192.0.2.2                           0
-           65542
Instance: default, AFI: ipv4, SAFI: vpn-unicast
  Hostname: PE2, Peer IP: 192.0.2.2
  Source IP: 192.0.2.1, Total routes: 2
    Prefix                                      Next Hop
MED         Lpref       AS Path
    198.51.100.221/32                           198.51.100.21                       0
-           65542
    198.51.100.222/32                           198.51.100.21                       0
-           65542
Instance: ce1-vrf, AFI: ipv6, SAFI: unicast
  Hostname: Local, Peer IP: ::
  Source IP: ::, Total routes: 2
    Prefix                                      Next Hop
MED         Lpref       AS Path
    2001:db8:0:1::201/128                       -                                   0
-           -
    2001:db8:0:1::202/128                       -                                   0
-           -
Instance: default, AFI: ipv6, SAFI: labeled-unicast
  Hostname: PE2, Peer IP: 192.0.2.2
  Source IP: 192.0.2.1, Total routes: 4
    Prefix                                      Next Hop
MED         Lpref       AS Path
    2001:db8:1:1::/64                           192.0.2.2                           0
-           65542
    2001:db8:2:1::/64                           192.0.2.2                           0
-           65542
    2001:db8:0:1::21/128                        192.0.2.2                           0
-           65542
    2001:db8:0:1::22/128                        192.0.2.2                           0
-           65542
```

```
Instance: default, AFI: ipv6, SAFI: labeled-unicast
  Hostname: Local, Peer IP: ::
  Source IP: ::, Total routes: 4
    Prefix                                      Next Hop
MED         Lpref       AS Path
    2001:db8:1:1::/64                           -                                   0
-           -
    2001:db8:2:1::/64                           -                                   0
-           -
    2001:db8:0:1::11/128                        -                                   0
-           -
    2001:db8:0:1::12/128                        -                                   0
-           -
Instance: default, AFI: ipv6, SAFI: vpn-unicast
  Hostname: PE2, Peer IP: 192.0.2.2
  Source IP: 192.0.2.1, Total routes: 2
    Prefix                                      Next Hop
MED         Lpref       AS Path
    2001:db8:0:1::221/128                       198.51.100.21                       0
-           65542
    2001:db8:0:1::222/128                       198.51.100.21                       0
-           65542
Instance: ce1-vrf, AFI: l2vpn, SAFI: evpn-vpws
  Hostname: Local, Peer IP: 0.0.0.0
  Source IP: 0.0.0.0, Total routes: 2
    Prefix                                      Next Hop
MED         Lpref       AS Path
    00.00.00.00.00.00.00.00.00.00:100/112       -                                   0
-           -
    00.00.00.00.00.00.00.00.00.00:200/112       -                                   0
-           -
Instance: default, AFI: l2vpn, SAFI: evpn
  Hostname: PE2, Peer IP: 192.0.2.2
  Source IP: 192.0.2.1, Total routes: 2
    Prefix                                      Next Hop
MED         Lpref       AS Path
    00.00.00.00.00.00.00.00.00.00:101/112       198.51.100.21                       0
-           65542
    00.00.00.00.00.00.00.00.00.00:201/112       198.51.100.21                       0
-           65542
```

Example: Summary view of the BGP rib-in where you can find received information for address family `l2vpn vpws`.

```
supervisor@rtbrick: op> show bgp rib-in l2vpn evpn
Instance: default, AFI: l2vpn, SAFI: evpn
  Hostname: PE2, Peer IP: 192.0.2.2
  Source IP: 192.0.2.1, Total routes: 2
    Prefix                                      Next Hop
MED         Lpref       AS Path
```

```
    00.00.00.00.00.00.00.00.00.00:101/112        198.51.100.21                    0
-           65542
    00.00.00.00.00.00.00.00.00.00:201/112        198.51.100.21                    0
-           65542
```

#### BGP FIB

The 'show bgp fib' commands display the BGP forwarding table.

**Syntax:**

**show bgp fib** <option>

| Option | Description |
|---|---|
| - | Without any option, the commands display the BGP forwarding table for all address families and all instances in a summary table format. |
| <afi> | BGP forwarding table summary for the given address family (AFI), all sub-address families and all instances. Supported AFI values are l2vpn, 'ipv4' and 'ipv6'. |
| <afi> <safi> | BGP forwarding table summary for the given address family (AFI) and sub-address family (SAFI), and all instances. Supported SAFI values are 'unicast', 'labeled-unicast', 'vpn-multicast', 'vpn-unicast', and evpn-vpws. |
| <afi> <safi> detail | Detailed list view of the BGP forwarding table for the given address family (AFI) and sub-address family (SAFI), and all instances. |
| <afi> <safi> <prefix> | BGP forwarding table entry for the given prefix and all instances. |
| <afi> <safi> instance <instance-name> | BGP forwarding table summary for the given AFI, SAFI, and instance. |
| <afi> <safi> instance <instance-name> detail | Detailed list view of BGP forwarding table for the given AFI, SAFI, and instance. |
| <afi> <safi> instance <instance-name> <prefix> | BGP forwarding table entry for the given prefix and instance. |

Example: Summary view of the BGP FIB

```
supervisor@rtbrick: op> show bgp fib
Instance: ce1-vrf, AFI: ipv4, SAFI: unicast, Total routes: 2
  Prefix                                        Preference      Label
Next Hop
  198.51.100.221/32                             20              519938,bos:1
198.51.100.21
  198.51.100.222/32                             20              519938,bos:1
```

```
198.51.100.21
Instance: default, AFI: ipv4, SAFI: labeled-unicast, Total routes: 2
  Prefix                                      Preference      Label
Next Hop
  198.51.100.21/32                            20              -
192.0.2.2
  198.51.100.22/32                            20              -
192.0.2.2
Instance: default, AFI: ipv4, SAFI: vpn-unicast, Total routes: 4
  Prefix                                      Preference      Label
Next Hop
  198.51.100.201/32                           200             519938,bos:1         -
  198.51.100.202/32                           200             519938,bos:1         -
  198.51.100.221/32                           20              519938,bos:1
198.51.100.21
  198.51.100.222/32                           20              519938,bos:1
198.51.100.21
Instance: ce1-vrf, AFI: ipv6, SAFI: unicast, Total routes: 2
  Prefix                                      Preference      Label
Next Hop
  2001:db8:0:1::221/128                       20              519939,bos:1
198.51.100.21
  2001:db8:0:1::222/128                       20              519939,bos:1
198.51.100.21
Instance: default, AFI: ipv6, SAFI: labeled-unicast, Total routes: 2
  Prefix                                      Preference      Label
Next Hop
  2001:db8:0:1::21/128                        20              -
192.0.2.2
  2001:db8:0:1::22/128                        20              -
192.0.2.2
Instance: default, AFI: ipv6, SAFI: vpn-unicast, Total routes: 4
  Prefix                                      Preference      Label
Next Hop
  2001:db8:0:1::201/128                       200             519939,bos:1         -
  2001:db8:0:1::202/128                       200             519939,bos:1         -
  2001:db8:0:1::221/128                       20              519939,bos:1
198.51.100.21
  2001:db8:0:1::222/128                       20              519939,bos:1
198.51.100.21
Instance: ce1-vrf, AFI: l2vpn, SAFI: evpn-vpws, Total routes: 2
  Prefix                                      Preference      Label
Next Hop
  00.00.00.00.00.00.00.00.00.00:101/112       20              519943,bos:1
198.51.100.21
  00.00.00.00.00.00.00.00.00.00:201/112       20              519944,bos:1
198.51.100.21
Instance: default, AFI: l2vpn, SAFI: evpn, Total routes: 4
  Prefix                                      Preference      Label
Next Hop
  00.00.00.00.00.00.00.00.00.00:100/112       200             519943,bos:1         -
```

```
  00.00.00.00.00.00.00.00.00.00:101/112         20              519943,bos:1
198.51.100.21
  00.00.00.00.00.00.00.00.00.00:200/112         200             519944,bos:1         -
  00.00.00.00.00.00.00.00.00.00:201/112         20              519944,bos:1
198.51.100.21
```

#### BGP RIB-out

This command displays the routes that were advertised to peers.

**Syntax:**

**show bgp rib-out** <option>

| Option | Description |
|---|---|
| - | Without any option, the command displays advertised BGP routes for all instances. |
| <afi> | BGP routing table summary for the given address family (AFI), all sub-address families and all instances. Supported AFI values are l2vpn, 'ipv4' and 'ipv6'. |
| <afi> <safi> | BGP routing table summary for the given address family (AFI) and sub-address family (SAFI), and all instances. Supported SAFI values are evpn, 'unicast', 'labeled-unicast', 'multicast', and 'vpn-unicast'. |
| <afi> <safi> detail | Detailed list view of the BGP routing table for the given address family (AFI) and sub-address family (SAFI), and all instances. |
| <afi> <safi> <prefix> | BGP routing table entry for the given prefix and all instances. |
| <afi> <safi> instance <instance-name> | BGP routing table summary for the given AFI, SAFI, and instance. |
| <afi> <safi> instance <instance-name> detail | Detailed list view of BGP routing table for the given AFI, SAFI, and instance. |
| <afi> <safi> instance <instance-name> <prefix> | BGP routing table entry for the given prefix and instance. |
| <afi> <safi> peer <name> / peer address <ip> | Peer name or address |

Example: Summary view of the routes advertised to a peer

```
supervisor@rtbrick: op> show bgp rib-out
Flags: N - RPKI Unknown, I - RPKI Invalid, V - RPKI Valid
Instance: default, AFI: ipv4, SAFI: labeled-unicast
  Peer-group: PE2, Sent routes: 6
    Flags  Prefix                                        MED     Lpref      Origin
Next Hop                                  AS Path
```

```
           192.0.2.0/30                                  0       -          Incomplete
-                                       65541
           192.0.2.4/30                                  0       -          Incomplete
-                                       65541
           198.51.100.11/32                              0       -          Incomplete
-                                       65541
           198.51.100.12/32                              0       -          Incomplete
-                                       65541
           198.51.100.21/32                              0       -          Incomplete
-                                       65541, 65542
           198.51.100.22/32                              0       -          Incomplete
-                                       65541, 65542
Instance: default, AFI: ipv4, SAFI: vpn-unicast
  Peer-group: PE2, Sent routes: 4
    Flags  Prefix                                        MED     Lpref      Origin
Next Hop                                AS Path
           198.51.100.201/32                             0       -          Incomplete
198.51.100.11                           65541
           198.51.100.202/32                             0       -          Incomplete
198.51.100.11                           65541
           198.51.100.221/32                             0       -          Incomplete
198.51.100.11                           65541, 65542
           198.51.100.222/32                             0       -          Incomplete
198.51.100.11                           65541, 65542
Instance: default, AFI: ipv6, SAFI: labeled-unicast
  Peer-group: PE2, Sent routes: 6
    Flags  Prefix                                        MED     Lpref      Origin
Next Hop                                AS Path
           2001:db8:1:1::/64                             0       -          Incomplete
-                                       65541
           2001:db8:2:1::/64                             0       -          Incomplete
-                                       65541
           2001:db8:0:1::11/128                          0       -          Incomplete
-                                       65541
           2001:db8:0:1::12/128                          0       -          Incomplete
-                                       65541
           2001:db8:0:1::21/128                          0       -          Incomplete
-                                       65541, 65542
           2001:db8:0:1::22/128                          0       -          Incomplete
-                                       65541, 65542
Instance: default, AFI: ipv6, SAFI: vpn-unicast
  Peer-group: PE2, Sent routes: 4
    Flags  Prefix                                        MED     Lpref      Origin
Next Hop                                AS Path
           2001:db8:0:1::201/128                         0       -          Incomplete
::ffff:198.51.100.11                    65541
           2001:db8:0:1::202/128                         0       -          Incomplete
::ffff:198.51.100.11                    65541
           2001:db8:0:1::221/128                         0       -          Incomplete
::ffff:198.51.100.11                    65541, 65542
           2001:db8:0:1::222/128                         0       -          Incomplete
```

```
::ffff:198.51.100.11                    65541, 65542
Instance: default, AFI: l2vpn, SAFI: evpn
  Peer-group: PE2, Sent routes: 4
    Flags  Prefix                                        MED     Lpref      Origin
Next Hop                                AS Path
           00.00.00.00.00.00.00.00.00.00:100/112         0       -          Incomplete
198.51.100.11                           65541
           00.00.00.00.00.00.00.00.00.00:101/112         0       -          Incomplete
198.51.100.11                           65541, 65542
           00.00.00.00.00.00.00.00.00.00:200/112         0       -          Incomplete
198.51.100.11                           65541
           00.00.00.00.00.00.00.00.00.00:201/112         0       -          Incomplete
198.51.100.11                           65541, 65542
```

#### BGP L2VPN

This command displays the BGP L2VPN information.

**Syntax:**

**show bgp l2vpn** <option>

| Option | Description |
|---|---|
| `pseudowire` | Displays pseudowire information for all instances. |
| `pseudowire instance <name>` | Displays pseudowire information for the specified instance. |

Example: Display L2VPN pseudowire information for instance `evpn-vpws-vrf1`

```
supervisor@rtbrick: op> show bgp l2vpn pseudowire instance ce1-vrf
Instance: ce1-vrf, AFI: l2vpn, SAFI: evpn-vpws
  Route distinguisher: 198.51.100.201:64501
  Number of local interfaces: 2
    Peer IP                                     Interface Name       Prefix
Status          Local SID  Remote SID  Pop Label          Push Label
    192.0.2.2                                   ifl-0/1/29/100
00.00.00.00.00.00.00.00.00.00:100/112      Up                    100         101
label:519943,bos:1  label:519943,bos:1
    192.0.2.2                                   ifl-0/1/29/200
00.00.00.00.00.00.00.00.00.00:200/112      Up                    200         201
label:519944,bos:1  label:519944,bos:1
```

# 3.2. BGP-signaled L2VPN

## 3.2.1. BGP-signaled L2VPN Overview

The BGP-signaled L2VPN uses BGP for signaling and auto-discovery to establish multipoint Layer 2

VPN over the MPLS backbone network. The remote cross-connect is a point-to-point (P2P) service that connects two locations using the MPLS core network and MP-BGP. The remote cross-connect uses MPLS tunnels to traverse the backbone network. It offers a Layer 2 packet forwarding mode that connects access circuits (ACs). The RBFS implementation of BGP-signaled L2VPN is in accordance with RFC-6624 which supports L2VPN using BGP for auto-discovery and Signaling.

**Supported Standards**

| RFC Number | Description |
|---|---|
| 6624 | Layer 2 Virtual Private Networks Using BGP for Auto-Discovery and Signaling |

**BGP-signaled L2VPN Network Model**

As shown in the figure below, a BGP-signaled L2VPN network contains the following building blocks:

- Customer edge (CE)—Customer device directly connected to the service provider network.
- Provider edge (PE)—Service provider device connected to CEs. PEs provide access to the Pseudowire network and forward traffic between customer network sites by using public tunnels.
- Attachment circuit (AC)—A physical or virtual link between a CE and a PE.
- Pseudowire (PW)—A virtual bidirectional connection between two PEs. A PW comprises a pair of virtual links in opposite directions.
- MPLS transport tunnel—A connection that carries one or more PWs across the MPLS core or IP backbone, such as an MPLS tunnel.
- Cross-connect—A connection formed by two physical or virtual circuits, such as ACs and PWs, that switches packets between them.

The figure below shows the protocol packet exchange process in the BGP-signaled L2VPN.

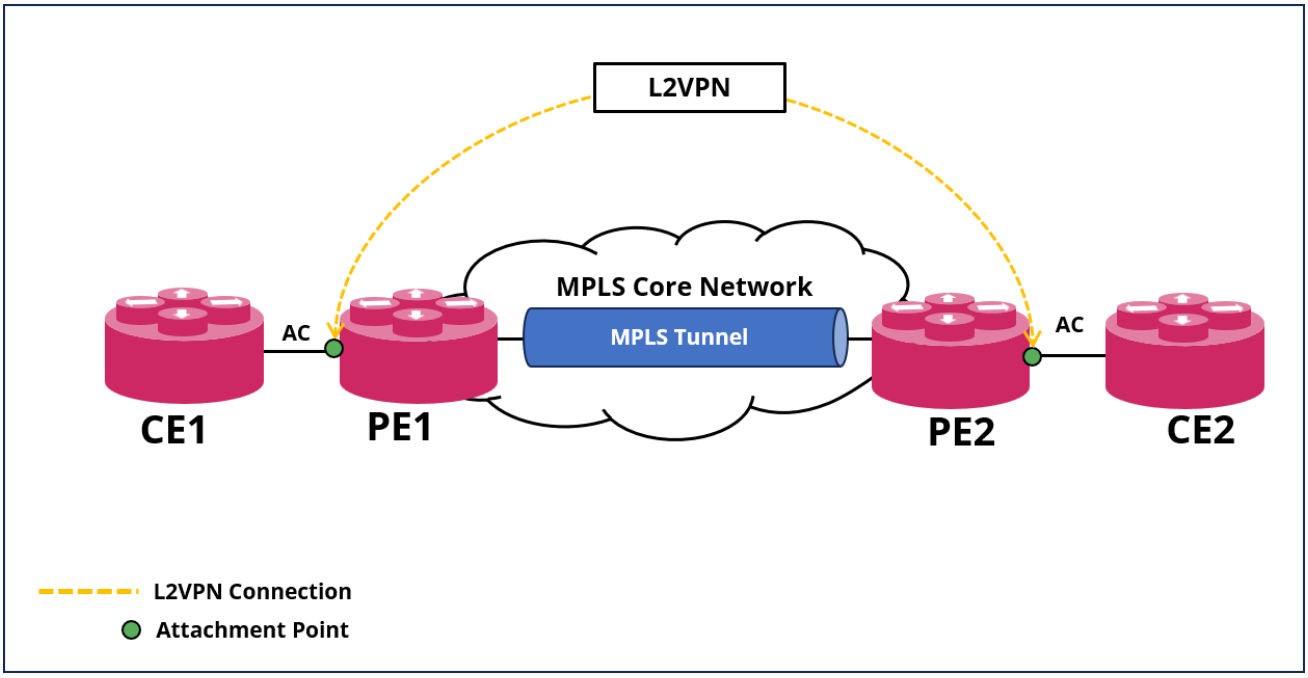



*Figure 46. Protocol Packet Exchange Process in the BGP-signaled L2VPN*

PE1 and PE2 are each configured with a BGP L2VPN instance. After the PE receives packets from the AC, it adds the PW label and sends them to the peer PE through the MPLS transport tunnel. After the other PE receives the packet via the MPLS transport path, it removes the PW label of the packets and forwards the packets to the AC bound to the PW.

### VLAN Tag Manipulation

VLAN Tag Manipulation involve modifying VLAN tags on Ethernet frames to ensure proper traffic segmentation and delivery across a Layer 2 network. RBFS supports some VLAN Tag Manipulation operations which are used to manipulate tags on Ethernet frames as they enter or leave a network interface. These manipulations include adding, removing, and rewriting VLAN tags. VLAN manipulation operations can be performed both at ingress and egress nodes.

The following VLAN manipulation operations are supported for Layer 2 VPN interfaces:

- **push**: Adds a single outer VLAN tag to an Ethernet frame.
- **pop**: Removes a single outer VLAN tag from an Ethernet tag frame.
- **swap**: Rewrite a single outer VLAN tag of an Ethernet tag frame.
- **push–push**: Adds two VLAN tags to an Ethernet frame.
- **pop-pop**: Removes two VLAN tags from an Ethernet tags frame.
- **swap–swap**: Rewrite both the outer and inner VLAN tags of an Ethernet tags frame.
- **swap–push**: Rewrite the outer VLAN tag and add a new inner VLAN tag for Ethernet tags' frame.
- **pop–swap**: Removes the outer VLAN tag and rewrites the inner VLAN tag of an Ethernet tags' frame.

### Supported CLI Options for VLAN Tag Manipulation Operations

The following table presents the various VLAN tag manipulation operations and supported CLI options.

| CLI Options | pop | push | swap | push-push | pop-pop | swap-push | swap-swap | pop-swap |
|---|---|---|---|---|---|---|---|---|
| vlan-id | No | Yes | Yes | Yes | No | Yes | Yes | Yes |
| vlan-encapsulation | No | Yes | Yes | Yes | No | Yes | Yes | Yes |
| innervlan-id | No | No | No | Yes | No | Yes | Yes | No |

#### Unsupported Features

The following features are not supported for the BGP-signaled L2VPN:

- Policy filtering options (at Address Family/Peer-Group/Instance levels)
- LAG interface as an attachment circuit
- LAG interface as an LSP MPLS path
- Route Reflection
- Add-Path
- VLAN change configuration on L2 IFL

#### Guidelines and Limitations

- The `control-word` flag is disabled by default in Layer 2 Attributes Extended Community. To ensure interoperability with RBFS, the `control-word` flag needs to be disabled.
- **LLDP protocol packet cross-connects via BGP L2VPN Pseudowire:** By default, LLDP protocol packets are cross-connected in the RtBrick stack and are not punted to the control plane.

#### Supported Platforms

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

### 3.2.2. Configuring BGP-signaled L2VPN

#### Configuration Hierarchy

The diagram illustrates the BGP-signaled L2VPN configuration hierarchy. All BGP-signaled L2VPN configurations are done within an instance, such as the default instance or an EVPN service instance.

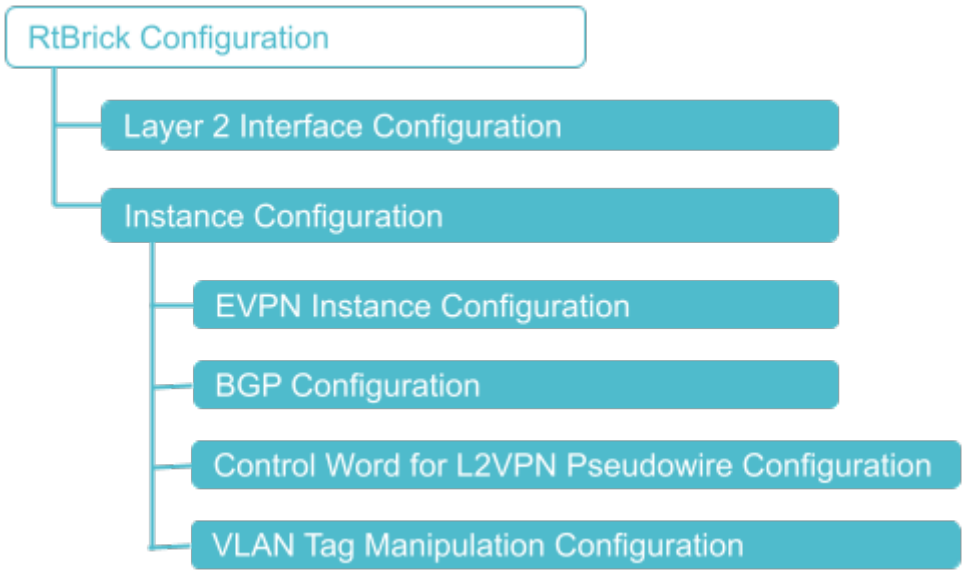



*Figure 47. BGP-signaled L2VPN Configuration Hierarchy*

### Configuration Syntax and Commands

The following sections describe the BGP-signaled L2VPN configuration syntax and commands.

#### Layer 2 Interface Configuration

BGP-signaled L2VPN supports the configuration of Layer 2 logical interfaces.

**Syntax:**

**set interface** <name> <attribute> <value>

| Attribute | Description |
|---|---|
| `<name>` | Specify the name of the interface. Example: ifp-0/0/1. |
| `unit <unit-id>` | Create a logical interface (also referred to as a sub-interface) under the physical interface |
| `unit <unit-id> interface-type l2vpn-vpws` | Specify the type of the L2VPN interface. |
| `unit <unit-id> match-type port` | Sets the "port" match-type for the L2VPN VPWS untagged interfaces. For tagged interfaces, "port" match-type is not supported. Ensure that there are no other tagged configurations for the interfaces when the port match-type is configured. When port matching is configured, it will match all traffic regardless of whether it is untagged, single-tagged, or double-tagged VLANs. |
| `unit <unit-id> vlan <outer-vlan-id>` | Outer VLAN ID. |
| `unit <unit-id> vlan <inner-vlan-id>` | Inner VLAN ID. |

| Attribute | Description |
| --- | --- |
| `unit <unit-id> instance <instance>` | Assign the logical interface to an instance. |

The following example shows an untagged interface configuration along with "port" match-type for the L2VPN VPWS untagged interface.

```
set interface ifp-0/0/17 unit 0
set interface ifp-0/0/17 unit 0 interface-type l2vpn-vpws
set interface ifp-0/0/17 unit 0 match-type port
set interface ifp-0/0/17 unit 0 instance evpn-vpws-vrf1
commit
```

The following example shows a single-tagged interface configuration:

```
set interface ifp-0/0/17 unit 100
set interface ifp-0/0/17 unit 100 interface-type l2vpn-vpws
set interface ifp-0/0/17 unit 100 instance l2vpn_ce1
set interface ifp-0/0/17 unit 100 vlan 100
commit
```

The following example shows a double-tagged interface configuration:

```
set interface ifp-0/0/17 unit 200
set interface ifp-0/0/17 unit 200 interface-type l2vpn-vpws
set interface ifp-0/0/17 unit 200 instance l2vpn_ce2
set interface ifp-0/0/17 unit 200 vlan 200
set interface ifp-0/0/17 unit 200 inner-vlan 201
commit
```

#### EVPN Instance Configuration

An EVPN Instance (EVI) is a routing and forwarding instance of EVPN that covers all the participating PE routers in a VPN. EVI is configured per customer on the PE routers. Each EVI has a unique route distinguisher and one or more route targets.

**Syntax:**

**set instance** <name> <attribute> <value>

| Attribute | Description |
| --- | --- |
| `<name>` | A unique name for the EVPN routing instance. |
| `address-family <afi>` | Address family identifier (AFI). Supported value: l2vpn |

| Attribute | Description |
|---|---|
| address-family <afi> <safi> | Subsequent address family identifier (SAFI). Supported values: vpls-vpws<br>The SAFI vpls-vpws needs to be enabled on the VRF instance. |
| protocol | Specifies the routing protocol |
| route-distinguisher <as-number \| ipv4-address:id> | The route distinguisher (RD) uniquely defines routes within an IPv4 network. PE routers use route distinguishers to identify which VPN a packet belongs to. Supported formats are <as-number:id> or <ipv4-address:id>. |
| ipv4-router-id <ipv4-router-id> | The router ID of the routing instance. |
| route-target ( import \| export ) <rt-value> | Route targets (RT) are used to transfer routes between VPN instances. The RT identifies a subset of routes that should be imported to or exported from a particular VPN instance. You can configure an RT for importing or exporting routes or both. |

In the following configuration, VRF instance AFI has been set to AFI: l2vpn and SAFI: vpls-vpws.

```
set instance l2vpn_ce1
set instance l2vpn_ce1 ipv4-router-id 182.1.6.4
set instance l2vpn_ce1 route-distinguisher 182.1.6.4:65006
set instance l2vpn_ce1 address-family l2vpn vpls-vpws
set instance l2vpn_ce1 address-family l2vpn vpls-vpws route-target import
target:182.1.6.0:65006
set instance l2vpn_ce1 address-family l2vpn vpls-vpws route-target export
target:182.1.6.0:65006
commit
```

#### BGP Configuration

#### BGP L2VPN VFT (Virtual Forwarding Table) Configuration

**Syntax:**

**set instance** <name> **protocol bgp** <attribute> <value>

| Attribute | Description |
|---|---|
| <name> | Name of the routing instance |
| address-family <afi> | Address family identifier (AFI). Supported value: l2vpn |

| Attribute | Description |
|---|---|
| `address-family <afi> <safi>` | Subsequent address family identifier (SAFI). Supported value: vpls-vpws<br>The SAFI `vpls-vpws` needs to be enabled on the VFT instance. |
| `local-as <as-number>` | The AS number in four-byte format. The numbers allowed are from 1 to 4294967295. |
| `interface <name>` | Interface that is bound to L2VPN |
| `nterface <name> local-site-id <local-site-id>` | Specify the local site ID that is used to establish an EVPN PW between two PEs |
| `interface <name> remote-site-id <remote-site-id>` | Specify the remote site ID that is used to establish an EVPN PW between two PEs |
| `interface <name> label-base <value>` | Specify the label base value. |

The following example configures `l2vpn_ce1` instance for BGP L2VPN.

```
set instance l2vpn_ce1 protocol bgp local-as 65006
set instance l2vpn_ce1 protocol bgp address-family l2vpn vpls-vpws
set instance l2vpn_ce1 protocol bgp address-family l2vpn vpls-vpws interface ifl-
0/0/17/100
set instance l2vpn_ce1 protocol bgp address-family l2vpn vpls-vpws interface ifl-
0/0/17/100 local-site-id 300
set instance l2vpn_ce1 protocol bgp address-family l2vpn vpls-vpws interface ifl-
0/0/17/100 remote-site-id 301
set instance l2vpn_ce1 protocol bgp address-family l2vpn vpls-vpws interface ifl-
0/0/17/100 label-base 150000
commit
```

#### BGP L2VPN Configuration

##### Configuring the BGP L2VPN Address Family

**Syntax:**

**set instance** <name> **protocol bgp address-family l2vpn vpls**

| Attribute | Description |
|---|---|
| `<name>` | Name of the routing instance |

To configure BGP L2VPN VPLS on the default instance, enter the following command:

```
set instance default protocol bgp address-family l2vpn vpls
```

```
commit
```

#### Configuring Address Families for Peer Groups

**Syntax:**

> **set instance** <instance-name> **protocol bgp peer-group** <pg-name> **address-family** <afi> <safi> <attribute> <value>

| Attribute | Description |
|---|---|
| `<afi>` | Address family identifier (AFI). Supported value: l2vpn. |
| `<safi>` | Subsequent address family identifier (SAFI). Supported value: vpls |
| `extended-nexthop` | Enable extended-next-hop encoding for BGP peer groups to allow the transfer of IPv4 prefixes over an IPv6 connection. |
| `update-nexthop ( ipv4-address \| ipv6-address ) <address>` | BGP nexthop address for routes advertised to this peer group |

To configure BGP L2VPN Peer Group on the `default` instance, enter the following command:

```
set instance default protocol bgp peer-group spine address-family l2vpn vpls
commit
```

#### Configuring Control Word for L2VPN Pseudowire

A control word can be inserted between the label stack and the MPLS payload. Control word is disabled by default.

**Syntax**:

> **set instance** <name> **protocol bgp address-family l2vpn vpls-vpws interface** <interface> **control-word enable**

| Attribute | Description |
|---|---|
| `<name>` | Name of the routing instance. |
| `<interface>` | Name of the interface. |

Example: Enter the following command to enable Control Word on the `default` instance.

```
set instance default protocol bgp peer-group spine address-family l2vpn vpls-vpws
interface ifl-0/0/1/10 control-word enable
commit
```

#### Configuring VLAN Tag Manipulation

The following section describes the VLAN tag manipulation configuration syntax and commands.

**Syntax:**

**set interface** <interface-name> **unit** <unit id> **[input-vlan-map** | **output-vlan-map]** <attribute> <value>

| Attribute | Description |
|---|---|
| `<name>` | Specify the name of the interface. |
| `unit <unit-id>` | Specify the unit ID for the interface. |
| `unit <unit-id> input-vlan-map` | Ingress VLAN mapping. |
| `unit <unit-id> output-vlan-map` | Egress VLAN mapping |
| `vlan-operation` | Specify any of the VLAN operations supported: `pop`, `pop-pop`, `pop-swap`, `push`, `push-push`, `swap`, `swap-push`, and `swap-swap`. |
| `vlan-encapsulation` | Specify the encapsulation type for the VLAN operation. Supported only 802.1q. |
| `vlan-id` | Specify the outer VLAN for the VLAN operation specified. |
| `inner-vlan-id` | Specify the inner VLAN for the VLAN operation specified. |

Example for VLAN operation 'push' configuration

```
supervisor@rtbrick.net: cfg> show config set interface ifp-0/1/4 unit 1000
set interface ifp-0/1/4 unit 1000
set interface ifp-0/1/4 unit 1000 interface-type l2vpn-vpws
set interface ifp-0/1/4 unit 1000 input-vlan-map vlan-operation push
set interface ifp-0/1/4 unit 1000 input-vlan-map vlan-id 1001
set interface ifp-0/1/4 unit 1000 instance l2vpn_ce1
set interface ifp-0/1/4 unit 1000 vlan 1000
```

Example: VLAN operation 'push' configuration output

```
supervisor@rtbrick.net: cfg> show config interface ifp-0/1/4 unit 1000
{
  "rtbrick-config:unit": [
    {
      "unit-id": 1000,
      "interface-type": "l2vpn-vpws",
      "input-vlan-map": {
        "vlan-operation": "push",
        "vlan-id": 1001
      },
      "instance": "l2vpn_ce1",
```

```
      "vlan": 1000
    }
  ]
}
```

Example for VLAN operation 'push-push' configuration

```
supervisor@rtbrick.net: cfg> show config set interface ifp-0/1/5 unit 2000
set interface ifp-0/1/5 unit 2000
set interface ifp-0/1/5 unit 2000 interface-type l2vpn-vpws
set interface ifp-0/1/5 unit 2000 input-vlan-map vlan-operation push-push
set interface ifp-0/1/5 unit 2000 input-vlan-map vlan-id 2000
set interface ifp-0/1/5 unit 2000 input-vlan-map inner-vlan-id 2001
set interface ifp-0/1/5 unit 2000 instance l2vpn_ce1
set interface ifp-0/1/5 unit 2000 vlan 2000
set interface ifp-0/1/5 unit 2000 inner-vlan 2
```

Example: VLAN operation 'push-push' configuration

```
supervisor@rtbrick.net: cfg> show config interface ifp-0/1/5 unit 2000
{
  "rtbrick-config:unit": [
    {
      "unit-id": 2000,
      "interface-type": "l2vpn-vpws",
      "input-vlan-map": {
        "vlan-operation": "push-push",
        "vlan-id": 2000,
        "inner-vlan-id": 2001
      },
      "instance": "l2vpn_ce1",
      "vlan": 2000,
      "inner-vlan": 2
    }
  ]
}
```

### 3.2.3. BGP L2VPN Operational Commands

#### BGP L2VPN Show Commands

The following show commands display the BGP L2VPN-related information.

##### BGP Summary

This command displays BGP protocol parameters like attributes or timers that are generic to the BGP instance.

**Syntax:**

**show bgp summary** <option>

| Attribute | Description |
|---|---|
| - | Without any option, the command displays the information for all instances. |
| instance <instance-name> | BGP summary information for the given instance. |

Example: BGP summary for instance l2vpn_ce1

```
supervisor@rtbrick: op> show bgp summary instance ce1-vrf
Instance: ce1-vrf
  General information
    Hostname: ce1-vrf instance, Domain name: ce1-vrf instance
    Local AS: 64501, Version: 4
    Local preference: 100, eBGP Protocol preference: 20, iBGP Protocol preference: 200
    Router ID: 198.51.100.201, Cluster ID: 198.51.100.201
    MED: 0, Enforce first AS: True
    Address families configured:
      ipv4: unicast
      ipv6: unicast
      l2vpn: vpls-vpws
  Capabilities
    Route refresh: True, AS4: True, Graceful restart: False, L2VPN VPLS-VPWS:True
  Best route selection
    Always compare MED: True, Ignore as path: False
    Ignore local preference: False, Ignore origin: False
    Ignore MED: False, Ignore route source: False
    Ignore router ID: False, Ignore uptime: True
    Ignore cluster length: False, Ignore peer IP: False
    Route select parameter: 0
  Timers
    Connect retry: 30s, Keepalive: 30s, Holdtime: 90s
  Statistics
    Peers configured: 0, Peers auto discovery: 0
      Peers in idle          : 0
      Peers in connect       : 0
      Peers in active        : 0
      Peers in opensent      : 0
      Peers in openconfirm   : 0
      Peers in established   : 0
```

**BGP Peer**

The 'show bgp peer' commands display information on BGP peers.

**Syntax:**

**show bgp peer** <option>

| Option | Description |
|---|---|
| - | Without any option, the commands display all BGP peers in all instances in a summary table format. |
| detail | Detailed information on all BGP peers in all instances in a list view. |
| <peer-name> | Detailed information on the peer with the given name. |
| address <peer-address> | Detailed information on the peer with the given IP address. |
| history | Displays BGP peer history information such as the peer state down reasons. |
| history <peer-address> | Displays BGP peer history information such as the peer state down reasons for a specified peer. |
| instance <instance-name> | Summary of all BGP peers in the given instance. |
| instance <instance-name> detail | Detailed information on all BGP peers in the given instance. |
| instance <instance-name> detail <peer-name> | Detailed information on the peer with the given name in the given instance. |
| instance <instance-name> detail address <peer-address> | Detailed information on the peer with the given IP address in the given instance. |
| statistics | Received and sent BGP prefixes per AFI/SAFI for all peers in all instances. |
| statistics peer <peer-name> | Received and sent BGP prefixes per AFI/SAFI for the peer with the given name. |
| statistics peer address <peer-address> | Received and sent BGP prefixes per AFI/SAFI for the peer with the given IP address. |
| statistics instance <instance-name> peer <peer-name> | Received and sent BGP prefixes per AFI/SAFI for the peer with the given name in the given instance. |
| statistics instance <instance-name> peer address <peer-address> | Received and sent BGP prefixes per AFI/SAFI for the peer with the given IP address in the given instance. |

Example 1: BGP Peer Information for instance `default`

```
supervisor@rtbrick: op> show bgp peer instance default
Instance: default
  Peer                                    Remote AS    State         Up/Down Time              PfxRcvd
PfxSent
  PE2                                     65542        Established   0d:00h:00m:12s            14                  24
```

Example 2: Detailed view of BGP Peer Information

```
supervisor@rtbrick: op> show bgp peer detail
Peer: PE2, Peer IP: 192.0.2.2, Remote AS: 65542, Local IP: 192.0.2.1, Local AS: 65541, Any AS: False
  Type: ebgp, State: Established, Up/Down Time: Wed Jul 15 19:02:42 GMT +0000 2026
  Discovered on interface: -
  Last transition: Wed Jul 15 19:02:42 GMT +0000 2026, Flap count: 0
  Peer ID        : 198.51.100.21, Local ID: 198.51.100.11
  Instance       : default, Peer group: PE2
  6PE enabled    : False
  TTL security   : False, TTL limit: -
  Authentication:
    ID: BGP_AUTH_HMAC-SHA-256-128, Algorithm: HMAC-SHA-256-128
  Timer values:
    Peer keepalive : 30s, Local keepalive: 30s
    Peer holddown  : 90s, Local holddown : 90s
    Connect retry  : 30s
  Timers:
    Connect retry timer : 0
    Keepalive timer     : expires in 10s 994697us
    Holddown timer      : expires in 1m 16s 623288us
  NLRIs:
    Sent           : ['l2vpn-vpls', 'ipv4-vpn-unicast', 'ipv6-vpn-unicast', 'ipv4-labeled-unicast', 'ipv6-labeled-
unicast']
    Received       : ['l2vpn-vpls', 'ipv4-vpn-unicast', 'ipv6-vpn-unicast', 'ipv4-labeled-unicast', 'ipv6-labeled-
unicast']
    Negotiated     : ['l2vpn-vpls', 'ipv4-vpn-unicast', 'ipv6-vpn-unicast', 'ipv4-labeled-unicast', 'ipv6-labeled-
unicast']
  Capabilities:
    Addpath sent                  : None
    Addpath received              : None
    Addpath negotiated            : None
    Extended nexthop sent         : None
    Extended nexthop received     : None
    Extended nexthop negotiated   : None
    Multiple labels sent          : None
    Multiple labels received      : None
    Multiple labels negotiated    : None
    Capabilities:
      Feature                      Sent            Received        Negotiated
      Route refresh                True            True            True
      4 byte AS                    True            True            True
      Graceful restart             False           False           False
      Link local only              False           False           False
  Prefix Limit:
  End of RIB:
    Address family                 Sent                                Received
    IPv4 unicast                   never                               never
    IPv4 labeled-unicast           Wed Jul 15 19:02:47 GMT +0000 2026  Wed Jul 15 19:02:47 GMT +0000 2026
    IPv6 unicast                   never                               never
    IPv6 labeled-unicast           Wed Jul 15 19:02:47 GMT +0000 2026  Wed Jul 15 19:02:47 GMT +0000 2026
    IPv4 VPN-unicast               Wed Jul 15 19:02:47 GMT +0000 2026  Wed Jul 15 19:02:47 GMT +0000 2026
    IPv6 VPN-unicast               Wed Jul 15 19:02:47 GMT +0000 2026  Wed Jul 15 19:02:47 GMT +0000 2026
    IPv4 flowspec                  never                               never
    IPv6 flowspec                  never                               never
    IPv4 VPN-multicast             never                               never
    L2VPN VPLS                     Wed Jul 15 19:02:47 GMT +0000 2026  Wed Jul 15 19:02:47 GMT +0000 2026
    L2VPN EVPN                     never                               never
  Message stats:
    Session stats:
      Direction              Open          Update       Keepalive          Notify   Route refresh
      Input                     1              11               1               0               0
      Output                    1              11               1               0               0
    Total stats:
      Input                     1              11               1               0               0
```

```
    Output                    1             11             1              0              0
  Route stats:
    Address family                        Received           Sent    Prefix limit    Idle timeout
    IPv4 unicast                                 0              0               0               0
    IPv4 labeled-unicast                         4              6               0               0
    IPv6 unicast                                 0              0               0               0
    IPv6 labeled-unicast                         4              6               0               0
    IPv4 VPN-unicast                             2              4               0               0
    IPv6 VPN-unicast                             2              4               0               0
    IPv4 multicast                               0              0               0               0
    IPv4 VPN-multicast                           0              0               0               0
    L2VPN VPLS                                   2              4               0               0
    L2VPN EVPN                                   0              0               0               0
    IPv4 Flowspec                                0              0               0               0
    IPv6 Flowspec                                0              0               0               0
```

Example 3: BGP Peer Statistics for a specific peer

```
supervisor@rtbrick: op> show bgp peer statistics peer address 192.0.2.2
Instance: default
  Peer                            AFI     SAFI                PfxRcvd    PfxSent
  PE2                             ipv4    unicast             0          0
                                  ipv4    labeled-unicast     4          6
                                  ipv6    unicast             0          0
                                  ipv6    labeled-unicast     4          6
                                  ipv4    vpn-unicast         2          4
                                  ipv6    vpn-unicast         2          4
                                  ipv4    multicast           0          0
                                  ipv4    vpn-multicast       0          0
                                  l2vpn   vpls                2          4
                                  l2vpn   evpn                0          0
                                  ipv4    flowspec            0          0
                                  ipv6    flowspec            0          0
```

Example 4: BGP Peer history for a specified peer

```
supervisor@rtbrick: op> show bgp peer history peer address 192.0.2.2
Instance: default
  Peer Address                            Source Address                          Type         Last Reset Reason
Update Time
  192.0.2.2                               192.0.2.1                               TCP Error    Cease, Sub-Cod...  Wed
Jul 15 19:02:12
  192.0.2.2                               192.0.2.1                               Config C...  Cease, Sub-Cod...  Wed
Jul 15 19:02:12
  192.0.2.2                               192.0.2.1                               Config C...  Cease, Sub-Cod...  Wed
Jul 15 19:01:24
  192.0.2.2                               192.0.2.1                               TCP Error    Cease, Sub-Cod...  Wed
Jul 15 19:00:33
  192.0.2.2                               192.0.2.1                               Config C...  Cease, Sub-Cod...  Wed
Jul 15 19:00:33
  192.0.2.2                               192.0.2.1                               Notifica...  Cease, Sub-Cod...  Wed
Jul 15 18:59:40
  192.0.2.2                               192.0.2.1                               Clear BG...  Cease, Sub-Cod...  Wed
Jul 15 18:59:37
```

#### BGP RIB-in

This command displays the received routes.

**Syntax:**

**show bgp rib-in** <option>

| Option | Description |
|---|---|
| - | Without any option, the command displays information on the received BGP routing table on all instances in a summary table format. |
| `<afi>` | BGP routing table summary for the given address family (AFI), all sub-address families and all instances. Supported AFI values are 'ipv4', 'ipv6' and `l2vpn`. |
| `<afi> <safi>` | BGP routing table summary for the given address family (AFI) and sub-address family (SAFI), and all instances. Supported SAFI values are `evpn`, 'labeled-unicast', 'unicast', 'vpn-multicast', 'vpn-unicast', and `vpls`. |
| `<afi> <safi> detail` | Detailed list view of the BGP routing table for the given address family (AFI) and sub-address family (SAFI), and all instances. |
| `<afi> <safi> <prefix>` | BGP routing table entry for the given prefix and all instances. |
| `<afi> <safi> instance <instance-name>` | BGP routing table summary for the given AFI, SAFI, and instance. |
| `<afi> <safi> instance <instance-name> detail` | Detailed list view of BGP routing table for the given AFI, SAFI, and instance. |
| `<afi> <safi> instance <instance-name> <prefix>` | BGP routing table entry for the given prefix and instance. |
| `<afi> <safi> community <community-name>` | BGP community details for the given AFI, SAFI, and instance. |
| `<afi> <safi> error` | BGP route with error status for the given AFI, SAFI, and instance. |
| `<afi> <safi> peer <name> / peer address <ip>` | Peer name or address |

Example 1: Summary view of the BGP rib-in where you can find received information for the BGP L2VPN instances like `l2vpn_ce1`, `l2vpn_ce2`, and `l2vpn_ce3`.

```
supervisor@rtbrick: op> show bgp rib-in
Instance: ce1-vrf, AFI: ipv4, SAFI: unicast
  Hostname: Local, Peer IP: 0.0.0.0
  Source IP: 0.0.0.0, Total routes: 2
    Prefix                                    Next Hop                        MED         Lpref       AS Path
    198.51.100.201/32                         -                               0           -           -
    198.51.100.202/32                         -                               0           -           -
Instance: default, AFI: ipv4, SAFI: labeled-unicast
  Hostname: Local, Peer IP: 0.0.0.0
  Source IP: 0.0.0.0, Total routes: 4
```

```
    Prefix                                    Next Hop                        MED        Lpref      AS Path
    192.0.2.0/30                              -                               0          -          -
    192.0.2.4/30                              -                               0          -          -
    198.51.100.11/32                          -                               0          -          -
    198.51.100.12/32                          -                               0          -          -
Instance: default, AFI: ipv4, SAFI: labeled-unicast
  Hostname: PE2, Peer IP: 192.0.2.2
  Source IP: 192.0.2.1, Total routes: 4
    Prefix                                    Next Hop                        MED        Lpref      AS Path
    192.0.2.0/30                              192.0.2.2                       0          -          65542
    192.0.2.4/30                              192.0.2.2                       0          -          65542
    198.51.100.21/32                          192.0.2.2                       0          -          65542
    198.51.100.22/32                          192.0.2.2                       0          -          65542
Instance: default, AFI: ipv4, SAFI: vpn-unicast
  Hostname: PE2, Peer IP: 192.0.2.2
  Source IP: 192.0.2.1, Total routes: 2
    Prefix                                    Next Hop                        MED        Lpref      AS Path
    198.51.100.221/32                         198.51.100.21                   0          -          65542
    198.51.100.222/32                         198.51.100.21                   0          -          65542
Instance: ce1-vrf, AFI: ipv6, SAFI: unicast
  Hostname: Local, Peer IP: ::
  Source IP: ::, Total routes: 2
    Prefix                                    Next Hop                        MED        Lpref      AS Path
    2001:db8:0:1::201/128                     -                               0          -          -
    2001:db8:0:1::202/128                     -                               0          -          -
Instance: default, AFI: ipv6, SAFI: labeled-unicast
  Hostname: PE2, Peer IP: 192.0.2.2
  Source IP: 192.0.2.1, Total routes: 4
    Prefix                                    Next Hop                        MED        Lpref      AS Path
    2001:db8:1:1::/64                         192.0.2.2                       0          -          65542
    2001:db8:2:1::/64                         192.0.2.2                       0          -          65542
    2001:db8:0:1::21/128                      192.0.2.2                       0          -          65542
    2001:db8:0:1::22/128                      192.0.2.2                       0          -          65542
Instance: default, AFI: ipv6, SAFI: labeled-unicast
  Hostname: Local, Peer IP: ::
  Source IP: ::, Total routes: 4
    Prefix                                    Next Hop                        MED        Lpref      AS Path
    2001:db8:1:1::/64                         -                               0          -          -
    2001:db8:2:1::/64                         -                               0          -          -
    2001:db8:0:1::11/128                      -                               0          -          -
    2001:db8:0:1::12/128                      -                               0          -          -
Instance: default, AFI: ipv6, SAFI: vpn-unicast
  Hostname: PE2, Peer IP: 192.0.2.2
  Source IP: 192.0.2.1, Total routes: 2
    Prefix                                    Next Hop                        MED        Lpref      AS Path
    2001:db8:0:1::221/128                     198.51.100.21                   0          -          65542
    2001:db8:0:1::222/128                     198.51.100.21                   0          -          65542
Instance: ce1-vrf, AFI: l2vpn, SAFI: vpls-vpws
  Hostname: Local, Peer IP: 0.0.0.0
  Source IP: 0.0.0.0, Total routes: 2
    Prefix                                    Next Hop                        MED        Lpref      AS Path
    00.00.00.00.00.00.00.00.00.00:2000/112    -                               0          -          -
    00.00.00.00.00.00.00.00.00.00:1000/112    -                               0          -          -
Instance: default, AFI: l2vpn, SAFI: vpls
  Hostname: PE2, Peer IP: 192.0.2.2
  Source IP: 192.0.2.1, Total routes: 2
    Prefix                                    Next Hop                        MED        Lpref      AS Path
    00.00.00.00.00.00.00.00.00.00:2001/112    198.51.100.21                   0          -          65542
    00.00.00.00.00.00.00.00.00.00:1001/112    198.51.100.21                   0          -          65542
```

Example 2: Summary view of the BGP rib-in where you can find received information for address family `l2vpn vpls`.

```
supervisor@rtbrick: op> show bgp rib-in l2vpn vpls
Instance: default, AFI: l2vpn, SAFI: vpls
  Hostname: PE2, Peer IP: 192.0.2.2
  Source IP: 192.0.2.1, Total routes: 2
    Prefix                                    Next Hop                        MED         Lpref       AS Path
    00.00.00.00.00.00.00.00.00.00:2001/112    198.51.100.21                   0           -           65542
    00.00.00.00.00.00.00.00.00.00:1001/112    198.51.100.21                   0           -           65542
```

#### BGP FIB

The 'show bgp fib' commands display the BGP forwarding table.

**Syntax:**

**show bgp fib** <option>

| Option | Description |
|---|---|
| - | Without any option, the commands display the BGP forwarding table for all address families and all instances in a summary table format. |
| <afi> | BGP forwarding table summary for the given address family (AFI), all sub-address families and all instances. Supported AFI values are l2vpn, 'ipv4' and 'ipv6'. |
| <afi> <safi> | BGP forwarding table summary for the given address family (AFI) and sub-address family (SAFI), and all instances. Supported SAFI values are 'unicast', 'labeled-unicast', 'vpn-multicast', 'vpn-unicast', 'evpn-vpws', and 'vpls-vpws'. |
| <afi> <safi> detail | Detailed list view of the BGP forwarding table for the given address family (AFI) and sub-address family (SAFI), and all instances. |
| <afi> <safi> <prefix> | BGP forwarding table entry for the given prefix and all instances. |
| <afi> <safi> instance <instance-name> | BGP forwarding table summary for the given AFI, SAFI, and instance. |
| <afi> <safi> instance <instance-name> detail | Detailed list view of BGP forwarding table for the given AFI, SAFI, and instance. |
| <afi> <safi> instance <instance-name> <prefix> | BGP forwarding table entry for the given prefix and instance. |

Example: Summary view of the BGP FIB

```
supervisor@rtbrick: op> show bgp fib
Instance: ce1-vrf, AFI: ipv4, SAFI: unicast, Total routes: 2
  Prefix                                      Preference      Label                Next Hop
  198.51.100.221/32                           20              519938,bos:1         198.51.100.21
  198.51.100.222/32                           20              519938,bos:1         198.51.100.21
```

```
Instance: default, AFI: ipv4, SAFI: labeled-unicast, Total routes: 2
  Prefix                                      Preference      Label                Next Hop
  198.51.100.21/32                            20              -                    192.0.2.2
  198.51.100.22/32                            20              -                    192.0.2.2
Instance: default, AFI: ipv4, SAFI: vpn-unicast, Total routes: 4
  Prefix                                      Preference      Label                Next Hop
  198.51.100.201/32                           200             519938,bos:1         -
  198.51.100.202/32                           200             519938,bos:1         -
  198.51.100.221/32                           20              519938,bos:1         198.51.100.21
  198.51.100.222/32                           20              519938,bos:1         198.51.100.21
Instance: ce1-vrf, AFI: ipv6, SAFI: unicast, Total routes: 2
  Prefix                                      Preference      Label                Next Hop
  2001:db8:0:1::221/128                       20              519939,bos:1         198.51.100.21
  2001:db8:0:1::222/128                       20              519939,bos:1         198.51.100.21
Instance: default, AFI: ipv6, SAFI: labeled-unicast, Total routes: 2
  Prefix                                      Preference      Label                Next Hop
  2001:db8:0:1::21/128                        20              -                    192.0.2.2
  2001:db8:0:1::22/128                        20              -                    192.0.2.2
Instance: default, AFI: ipv6, SAFI: vpn-unicast, Total routes: 4
  Prefix                                      Preference      Label                Next Hop
  2001:db8:0:1::201/128                       200             519939,bos:1         -
  2001:db8:0:1::202/128                       200             519939,bos:1         -
  2001:db8:0:1::221/128                       20              519939,bos:1         198.51.100.21
  2001:db8:0:1::222/128                       20              519939,bos:1         198.51.100.21
Instance: ce1-vrf, AFI: l2vpn, SAFI: vpls-vpws, Total routes: 2
  Prefix                                      Preference      Label                Next Hop
  00.00.00.00.00.00.00.00.00.00:2001/112      20              202002               198.51.100.21
  00.00.00.00.00.00.00.00.00.00:1001/112      20              201002               198.51.100.21
Instance: default, AFI: l2vpn, SAFI: vpls, Total routes: 4
  Prefix                                      Preference      Label                Next Hop
  00.00.00.00.00.00.00.00.00.00:2000/112      200             202001,bos:1         -
  00.00.00.00.00.00.00.00.00.00:2001/112      20              202002               198.51.100.21
  00.00.00.00.00.00.00.00.00.00:1000/112      200             201001,bos:1         -
  00.00.00.00.00.00.00.00.00.00:1001/112      20              201002               198.51.100.21
```

#### BGP RIB-out

This command displays the routes that were advertised to peers.

**Syntax:**

**show bgp rib-out** <option>

| Option | Description |
|---|---|
| - | Without any option, the command displays advertised BGP routes for all instances. |
| <afi> | BGP routing table summary for the given address family (AFI), all sub-address families and all instances. Supported AFI values are l2vpn, 'ipv4' and 'ipv6'. |

| Option | Description |
|---|---|
| `<afi> <safi>` | BGP routing table summary for the given address family (AFI) and sub-address family (SAFI), and all instances. Supported SAFI values are `evpn`, 'unicast', 'labeled-unicast', 'multicast', 'vpn-unicast', and `vpls`. |
| `<afi> <safi> detail` | Detailed list view of the BGP routing table for the given address family (AFI) and sub-address family (SAFI), and all instances. |
| `<afi> <safi> <prefix>` | BGP routing table entry for the given prefix and all instances. |
| `<afi> <safi> instance <instance-name>` | BGP routing table summary for the given AFI, SAFI, and instance. |
| `<afi> <safi> instance <instance-name> detail` | Detailed list view of BGP routing table for the given AFI, SAFI, and instance. |
| `<afi> <safi> instance <instance-name> <prefix>` | BGP routing table entry for the given prefix and instance. |
| `<afi> <safi> peer <name> / peer address <ip>` | Peer name or address |

Example: Summary view of the routes advertised to a peer

```
supervisor@rtbrick: op> show bgp rib-out
Flags: N - RPKI Unknown, I - RPKI Invalid, V - RPKI Valid
Instance: default, AFI: ipv4, SAFI: labeled-unicast
  Peer-group: PE2, Sent routes: 6
    Flags  Prefix                                      MED     Lpref      Origin          Next Hop
AS Path
           192.0.2.0/30                                0       -          Incomplete      -
65541
           192.0.2.4/30                                0       -          Incomplete      -
65541
           198.51.100.11/32                            0       -          Incomplete      -
65541
           198.51.100.12/32                            0       -          Incomplete      -
65541
           198.51.100.21/32                            0       -          Incomplete      -
65541, 65542
           198.51.100.22/32                            0       -          Incomplete      -
65541, 65542
Instance: default, AFI: ipv4, SAFI: vpn-unicast
  Peer-group: PE2, Sent routes: 4
    Flags  Prefix                                      MED     Lpref      Origin          Next Hop
AS Path
           198.51.100.201/32                           0       -          Incomplete      198.51.100.11
65541
           198.51.100.202/32                           0       -          Incomplete      198.51.100.11
65541
           198.51.100.221/32                           0       -          Incomplete      198.51.100.11
65541, 65542
           198.51.100.222/32                           0       -          Incomplete      198.51.100.11
65541, 65542
Instance: default, AFI: ipv6, SAFI: labeled-unicast
  Peer-group: PE2, Sent routes: 6
    Flags  Prefix                                      MED     Lpref      Origin          Next Hop
AS Path
           2001:db8:1:1::/64                           0       -          Incomplete      -
65541
           2001:db8:2:1::/64                           0       -          Incomplete      -
```

```
65541
          2001:db8:0:1::11/128                        0       -          Incomplete      -
65541
          2001:db8:0:1::12/128                        0       -          Incomplete      -
65541
          2001:db8:0:1::21/128                        0       -          Incomplete      -
65541, 65542
          2001:db8:0:1::22/128                        0       -          Incomplete      -
65541, 65542
Instance: default, AFI: ipv6, SAFI: vpn-unicast
  Peer-group: PE2, Sent routes: 4
    Flags  Prefix                                     MED     Lpref      Origin          Next Hop
AS Path
          2001:db8:0:1::201/128                       0       -          Incomplete      ::ffff:198.51.100.11
65541
          2001:db8:0:1::202/128                       0       -          Incomplete      ::ffff:198.51.100.11
65541
          2001:db8:0:1::221/128                       0       -          Incomplete      ::ffff:198.51.100.11
65541, 65542
          2001:db8:0:1::222/128                       0       -          Incomplete      ::ffff:198.51.100.11
65541, 65542
Instance: default, AFI: l2vpn, SAFI: vpls
  Peer-group: PE2, Sent routes: 4
    Flags  Prefix                                     MED     Lpref      Origin          Next Hop
AS Path
          00.00.00.00.00.00.00.00.00.00:2000/112      0       -          Incomplete      198.51.100.11
65541
          00.00.00.00.00.00.00.00.00.00:2001/112      0       -          Incomplete      198.51.100.11
65541, 65542
          00.00.00.00.00.00.00.00.00.00:1000/112      0       -          Incomplete      198.51.100.11
65541
          00.00.00.00.00.00.00.00.00.00:1001/112      0       -          Incomplete      198.51.100.11
65541, 65542
```

#### BGP L2VPN

This command displays the BGP L2VPN information.

**Syntax:**

**show bgp l2vpn** <option>

| Option | Description |
|---|---|
| `pseudowire` | Displays pseudowire peer IP information for all instances. |
| `pseudowire instance <name>` | Displays pseudowire peer IP information for the specified instance. |
| pseudowire detail | Displays detailed pseudowire peer IP information for all instances. |

Example 1: Display L2VPN pseudowire information for instance `l2vpn_ce1`

```
supervisor@rtbrick: op> show bgp l2vpn pseudowire instance ce1-vrf
Instance: ce1-vrf, AFI: l2vpn, SAFI: vpls-vpws
  Route distinguisher: 198.51.100.201:64501
  Number of local interfaces: 2
    Peer IP                                   Interface Name       Prefix                                   Status
```

```
Local SID  Remote SID  Pop Label           Push Label
    192.0.2.2                                   ifl-0/1/29/1000      00.00.00.00.00.00.00.00.00.00:1000/112     Up
1000        1001  label:201001,bos:1  label:201002
    192.0.2.2                                   ifl-0/1/29/2000      00.00.00.00.00.00.00.00.00.00:2000/112     Up
2000        2001  label:202001,bos:1  label:202002
```

Example 2: Display L2VPN pseudowire information

```
supervisor@rtbrick: op> show bgp l2vpn pseudowire
Instance: ce1-vrf, AFI: l2vpn, SAFI: vpls-vpws
  Route distinguisher: 198.51.100.201:64501
  Number of local interfaces: 2
    Peer IP                                     Interface Name       Prefix                                     Status
Local SID  Remote SID  Pop Label           Push Label
    192.0.2.2                                   ifl-0/1/29/1000      00.00.00.00.00.00.00.00.00.00:1000/112     Up
1000        1001  label:201001,bos:1  label:201002
    192.0.2.2                                   ifl-0/1/29/2000      00.00.00.00.00.00.00.00.00.00:2000/112     Up
2000        2001  label:202001,bos:1  label:202002
```

Example 3: Detailed view of L2VPN pseudowire information

```
supervisor@rtbrick: op> show bgp l2vpn pseudowire detail
Instance: ce1-vrf, AFI: l2vpn, SAFI: vpls-vpws
  Route distinguisher: 198.51.100.201:64501, Number of local interfaces: 2
    Prefix: 00.00.00.00.00.00.00.00.00.00:1000/112
      Peer IP:  192.0.2.2, Interface name: ifl-0/1/29/1000, Status: Up
      Local SID: 1000, Remote SID: 1000, Pseudowire type: Remote cross connect
      Pop label: label:201001,bos:1, Push label: label:201002, Control Word: False
    Prefix: 00.00.00.00.00.00.00.00.00.00:2000/112
      Peer IP:  192.0.2.2, Interface name: ifl-0/1/29/2000, Status: Up
      Local SID: 2000, Remote SID: 2000, Pseudowire type: Remote cross connect
      Pop label: label:202001,bos:1, Push label: label:202002, Control Word: False
```

# 3.3. LDP-signaled L2VPN

## 3.3.1. LDP-signaled L2VPN Overview

The LDP-signaled L2VPN uses Targeted Label Distribution Protocol (T-LDP) for signaling to establish Pseudowire over the MPLS backbone network. The Pseudowire is a point-to-point (P2P) service that connects two locations using the MPLS core network and uses T-LDP for signaling. The Pseudowire uses MPLS tunnels to traverse the backbone network. It offers a Layer 2 packet forwarding mode that connects attachment circuits (ACs). The RBFS implementation of LDP-signaled L2VPN is in accordance with RFC 4447 which supports L2VPN using LDP for signaling.

**Supported Standards**

| RFC Number | Description |
|---|---|
| 4447 | Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP) |

RFC and draft compliance are partial except as specified.

**LDP-signaled L2VPN Network Model**

As shown in the figure below, a LDP-signaled L2VPN network contains the following building blocks:

- Customer edge (CE)—Customer device directly connected to the service provider network.
- Provider edge (PE)—Service provider device connected to CEs. PEs provide access to the Pseudowire network and forward traffic between customer network sites by using public tunnels.
- Attachment circuit (AC)—A physical or virtual link between a CE and a PE.
- Pseudowire (PW)—A virtual bidirectional connection between two PEs. A PW comprises a pair of virtual links in opposite directions.
- MPLS transport tunnel—A connection that carries one or more PWs across the MPLS core or IP backbone, such as an MPLS tunnel.
- Cross-connect—A connection formed by two physical or virtual circuits, such as ACs and PWs, that switches packets between them.

The figure below shows the protocol packet exchange process in the LDP-signaled L2VPN.

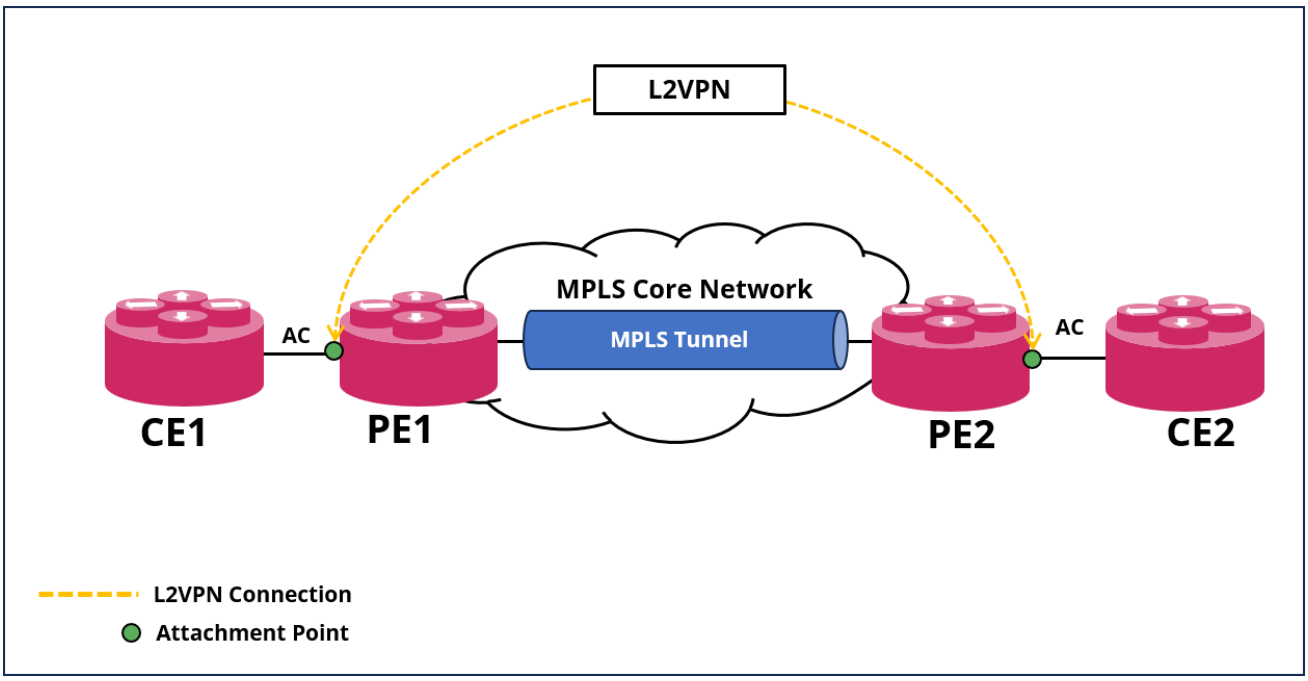



*Figure 48. Protocol Packet Exchange Process in the LDP-signaled L2VPN*

PE1 and PE2 are each configured with a LDP L2VPN instance with T-LDP neighbors to carry the Pseudowire (PW) tunnel information. After the PE receives packets from the AC, it adds the PW label and sends them to the peer PE through the MPLS transport tunnel. After the other PE receives the packet via the MPLS transport path, it removes the PW label of the packets and forwards the packets to the AC bound to the PW.

#### Pseudowire Redundancy

Pseudowire redundancy is supported only for the Provider Edge (PE) routers.

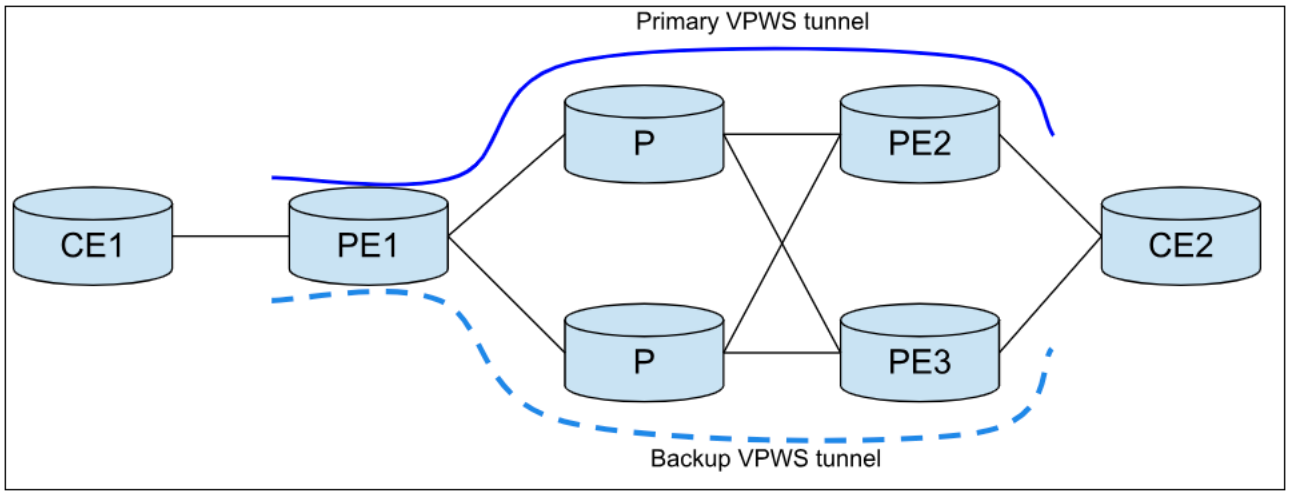



*Figure 49. Pseudowire Redundancy*

As shown in the figure above, PE1 has two VPWS tunnels. The tunnel from PE1 to PE2 is the primary tunnel, while the tunnel from PE1 to PE3 serves as the backup tunnel. Switching back to the primary pseudowire is not performed even if it recovers.

In the scenario with PE1 and PE2, where P3 acts as redundancy as shown in the diagram above, switching switching is exclusively executed by PE1 when certain conditions are met:

- The LDP session with the primary remote PE is down.
- A label withdrawal occurs from PE2.
- The MPLS tunnel to PE2 is down.
- The "request switch-over ldp l2vpn" operational command can be entered to switch to the backup PE. For details, see the request switch-over section.

#### Guidelines and Limitations

- The `control-word` flag is disabled by default, which can be enabled if required. For more details, see the Configuring LDP-signaled L2VPN section.

#### Supported Platforms

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform

### 3.3.2. Configuring LDP-signaled L2VPN

#### Configuration Hierarchy

The diagram illustrates the LDP-signaled L2VPN configuration hierarchy. All LDP-signaled L2VPN configurations are done within an instance, such as the default instance or VRF instance.

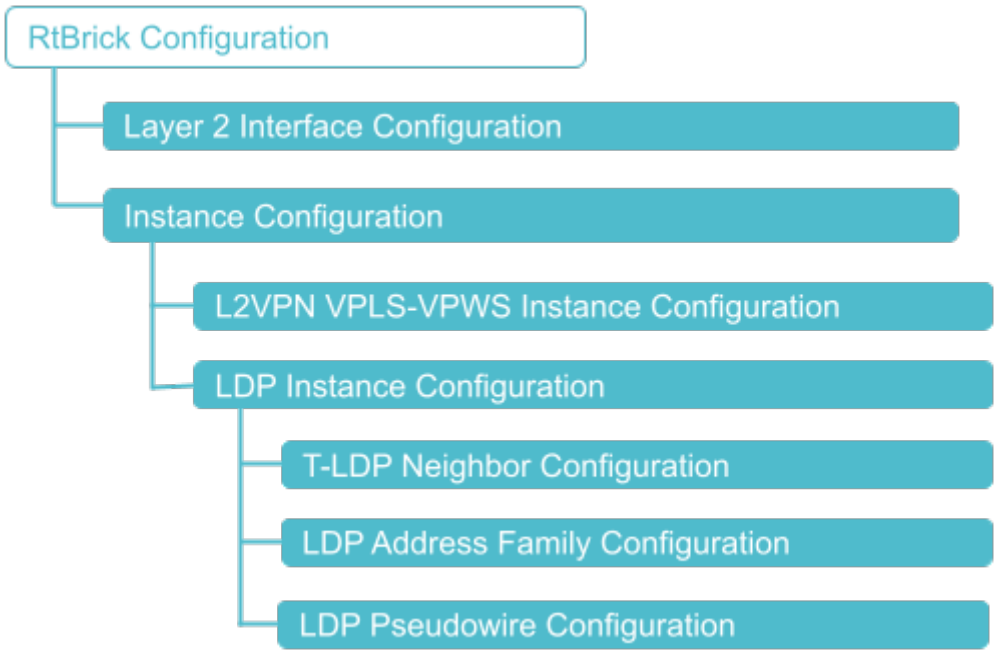



*Figure 50. LDP-signaled L2VPN Configuration Hierarchy*

### Configuration Syntax and Commands

The following sections describe the LDP-signaled L2VPN configuration syntax and commands.

#### Layer 2 Interface Configuration

LDP-signaled L2VPN supports the configuration of Layer 2 logical interfaces.

**Syntax:**

```
set interface <name> <attribute> <value>
```

| Attribute | Description |
|---|---|
| `<name>` | Specify the name of the interface. Example: ifp-0/0/1. |
| `unit <unit-id>` | Create a logical interface (also referred to as a sub-interface) under the physical interface |
| `unit <unit-id> interface-type l2vpn-vpws` | Specify the type of the L2VPN interface. |
| `unit <unit-id> vlan <outer-vlan-id>` | Outer VLAN ID. |
| `unit <unit-id> vlan <inner-vlan-id>` | Inner VLAN ID. |
| `unit <unit-id> instance <instance>` | Assign the logical interface to an instance. |

The following example shows a single-tagged interface configuration:

```
set interface ifp-0/0/17 unit 100
```

```
set interface ifp-0/0/17 unit 100 interface-type l2vpn-vpws
set interface ifp-0/0/17 unit 100 instance default
set interface ifp-0/0/17 unit 100 vlan 100
commit
```

The following example shows a double-tagged interface configuration:

```
set interface ifp-0/0/17 unit 200
set interface ifp-0/0/17 unit 200 interface-type l2vpn-vpws
set interface ifp-0/0/17 unit 200 instance default
set interface ifp-0/0/17 unit 200 vlan 200
set interface ifp-0/0/17 unit 200 inner-vlan 201
commit
```

#### Instance Configuration

##### L2VPN VPLS-VPWS Instance Configuration

This section explains how to enable the L2VPN `vpls-vpws` address family in the instance.

> By default, the SAFI `vpls-vpws` is disabled, which you should enable for L2VPN Pseudowire.

**Syntax:**

**set instance** <name> <attribute> <value>

| Attribute | Description |
|---|---|
| `<name>` | A unique name for the routing instance. |
| `address-family <afi>` | Address family identifier (AFI). Supported value: l2vpn |
| `address-family <afi> <safi>` | Subsequent address family identifier (SAFI). Supported values: vpls-vpws<br><br>The SAFI `vpls-vpws` needs to be enabled on the instance. |

Example: L2VPN VPLS-VPWS Instance Configuration

```
set instance default address-family l2vpn vpls-vpws
```

#### LDP Instance Configuration

##### LDP Targeted Neighbor Configuration

The targeted neighbor configuration establishes an LDP neighbor with a remote neighbor that is not directly connected.

**Syntax:**

**set instance** <instance> **protocol ldp targeted-neighbor** <attribute> <value>

| Attribute | Description |
|---|---|
| `ipv4 <neighbor-address>` | IPv4 address of the remote LDP neighbor. |
| `ipv4 <neighbor-address> <source-address>` | Neighbhor address to which ldp hello is sent |
| `ipv4 <neighbor-address> <source-address> hold-time <hold-time>` | Targeted hold time, <0-65535>, default 45s |
| `ipv4 <neighbor-address> <source-address> interval <interval>` | Targeted hello messages interval, <0-65535>, default 15s |
| `ipv4 <neighbor-address> <source-address> neighbor-type <receiver \|sender>` | Specifies the neighbor type, that is, sender or receiver. |
| `ipv6 <neighbor-address>` | IPv6 address of the remote LDP neighbor. |
| `ipv6 <neighbor-address> <source-address>` | Neighbhor address to which ldp hello is sent |
| `ipv6 <neighbor-address> <source-address> hold-time <hold-time>` | Targeted hold time, <0-65535>, default 45s |
| `ipv6 <neighbor-address> <source-address> interval <interval>` | Targeted hello messages interval, <0-65535>, default 15s |
| `ipv6 <neighbor-address> <source-address> neighbor-type <receiver \|sender>` | Specifies the neighbor type, that is, sender or receiver. |

```
set instance default protocol ldp targeted-neighbor ipv4 198.51.100.1 198.51.100.2
set instance default protocol ldp targeted-neighbor ipv4 198.51.100.5 198.51.100.2
```

Example: LDP Targeted Neighbor Configuration

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ldp targeted-
neighbor
{
  "rtbrick-config:targeted-neighbor": {
    "ipv4": [
      {
        "neighbor-address": "198.51.100.1",
        "source-address": "198.51.100.2"
      },
      {
        "neighbor-address": "198.51.100.5",
```

```
        "source-address": "198.51.100.2"
      }
    ]
  }
}
```

#### LDP Address Family Configuration

The address-family command allows you to enable the address families that LDP will route and configure settings that are specific to that address family.

> By default, the SAFI "vpls-vpws" is disabled, which you should enable for L2VPN Pseudowire.

**Syntax:**

**set instance** <instance-name> **protocol ldp address-family** <attribute> <value>

| Attribute | Description |
|---|---|
| `<afi>` | Address family identifier (AFI). Supported values: l2vpn |
| `<afi> <safi>` | Subsequent address family. Supported values: vpls-vpws |
| `<afi> <safi> status <enable\|disable>` | Enable or disable address family. |

Example 1: LDP Address Family Configuration for L2VPN Pseudowire

```
set instance default protocol ldp address-family l2vpn vpls-vpws status enable
```

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ldp address-
family
{
  "rtbrick-config:address-family": [
    {
      "afi": "l2vpn",
      "safi": "vpls-vpws",
      "status": "enable"
    }
  ]
}
```

#### LDP Pseudowire Configuration

**Syntax:**

**set instance** <instance-name> **protocol ldp vpls-vpws** <attribute> <value>

| Attribute | Description |
|---|---|
| `interface <interface>` | Attachment Circuit L2VPN Interface. |
| `interface <interface> pseudowire-id <pseudowire-id>` | Specifies the pseudowire identifier. |
| `interface <interface> neighbor <ipv4/ipv6> address <ipaddress>` | Specifies the T-LDP neighbor. |
| `interface <interface> backup-neighbor <ipv4/ipv6> address <ipaddress>` | Specifies the redundancy T-LDP neighbor. |
| `interface <interface> control-word <enable/disable>` | Specifies whether control-word is enabled or disabled. Default value: disable |

Example 1: LDP VPWS Configuration

```
set instance default protocol ldp vpls-vpws interface ifl-0/0/17/100
set instance default protocol ldp vpls-vpws interface ifl-0/0/17/100 pseudowire-id 100
set instance default protocol ldp vpls-vpws interface ifl-0/0/17/100 control-word
enable
set instance default protocol ldp vpls-vpws interface ifl-0/0/17/100 neighbor ipv4
address 192.51.100.1
```

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ldp vpls-vpws
interface ifl-0/0/17/100
{
  "rtbrick-config:interface": [
    {
      "name": "ifl-0/0/17/100",
      "pseudowire-id": 100,
      "control-word": "enable",
      "neighbor": {
        "ipv4": {
          "address": "192.51.100.1"
        }
      }
    }
  ]
}
```

Example 2: LDP VPWS Redundancy Configuration

```
set instance default protocol ldp vpls-vpws interface ifl-0/0/17/200
set instance default protocol ldp vpls-vpws interface ifl-0/0/17/200 pseudowire-id 200
```

```
set instance default protocol ldp vpls-vpws interface ifl-0/0/17/200 control-word
enable
set instance default protocol ldp vpls-vpws interface ifl-0/0/17/200 neighbor ipv4
address 192.51.100.1
set instance default protocol ldp vpls-vpws interface ifl-0/0/17/200 backup-neighbor
ipv4 address 192.51.100.5
```

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ldp vpls-vpws
interface ifl-0/0/17/200
{
  "rtbrick-config:interface": [
    {
      "name": "ifl-0/0/17/200",
      "pseudowire-id": 200,
      "control-word": "enable",
      "neighbor": {
        "ipv4": {
          "address": "192.51.100.1"
        }
      },
      "backup-neighbor": {
        "ipv4": {
          "address": "192.51.100.5"
        }
      }
    }
  ]
}
```

### 3.3.3. LDP L2VPN Operational Commands

**show ldp binding**

**Syntax:**

**show ldp binding** <option>

| Option | Description |
|---|---|
| - | Without any option, this command displays a summary of all the LDP label bindings. |
| detail | Displays detailed LDP label binding information. |
| instance <instance-name> | Displays LDP label binding information of the specified instance. |
| instance <instance-name> <afi> | Displays LDP label binding information of the specified address family. Supported AFI value: l2vpn. |

| Option | Description |
| --- | --- |
| `instance <instance-name> <afi> detail` | Displays detailed LDP label binding information of the specified address family. |
| `instance <instance-name> <afi> prefix <ip>` | Displays LDP label binding information of the specified prefix and instance for the specified address family. |
| `instance <instance-name> <afi> prefix <ip> detail` | Displays detailed LDP label binding information of the specified prefix and instance for the specified address family. |
| `received` | Displays the LDP received label binding information of the LDP sessions. |
| `received instance <instance-name> <afi>` | Displays LDP received label binding information of the specified address family and instance. |
| `received instance <instance-name> <afi> detail` | Displays detailed LDP received label binding information of the specified address family and instance. |
| `received instance <instance-name> <afi> ldp-id <ldp-id>` | Displays LDP received label binding information of the specified instance. |
| `received instance <instance-name> <afi> ldp-id <ldp-id> detail` | Displays detailed LDP received label binding information of the specified LDP identifier and instance. |
| `received <afi>` | Displays LDP received label binding information of the specified address family. Supported AFI values: l2vpn. |
| `received <afi> detail` | Displays detailed LDP received label binding information of the specified address family. |
| `received <afi> instance <instance-name>` | Displays LDP received label binding information of the specified address family and instance. |
| `received <afi> instance <instance-name> detail` | Displays detailed LDP received label binding information of the specified address family and instance. |
| `received <afi> instance <instance-name> ldp-id <ldp-id>` | Displays LDP received label binding information of the specified address family, instance, and LDP identifier. |
| `received <afi> ldp-id <ldp-id>` | Displays LDP received label binding information of the specified address family and LDP identifier. |
| `received <afi> ldp-id <ldp-id> detail` | Displays detailed LDP received label binding information of the specified address family and LDP identifier. |
| `sent` | Displays the LDP sent label binding information of the LDP sessions. |
| `sent instance <instance-name> <afi>` | Displays LDP sent label binding information of the specified address family and instance. Supported AFI values: l2vpn. |
| `sent instance <instance-name> <afi> detail` | Displays detailed LDP sent label binding information of the specified address family and instance. |
| `sent instance <instance-name> <afi> prefix <ip>` | Displays LDP sent label binding information of the specified address family, instance, and prefix. |

| Option | Description |
|---|---|
| `sent instance <instance-name> <afi> prefix <ip> detail` | Displays detailed LDP sent label binding information of the specified address family, instance, and prefix. |
| `sent <afi>` | Displays LDP sent label binding information of the specified address family. Supported AFI values: l2vpn. |
| `sent <afi> detail` | Displays detailed LDP sent label binding information of the specified address family. |
| `sent <afi> instance <instance-name>` | Displays LDP sent label binding information of the specified address family and instance. |
| `sent <afi> instance <instance-name> detail` | Displays detailed LDP sent label binding information of the specified address family and instance. |
| `sent <afi> instance <instance-name> prefix <ip>` | Displays LDP sent label binding information of the specified address family, instance, and prefix. |
| `sent <afi> instance <instance-name> prefix <ip> detail` | Displays detailed LDP sent label binding information of the specified address family, instance, and prefix |
| `sent <afi> prefix <ip>` | Displays LDP sent label binding information of the specified address family and prefix. |
| `sent <afi> prefix <ip> detail` | Displays detailed LDP sent label binding information of the specified address family and prefix. |

Example: Summary view of the LDP binding

```
supervisor@rtbrick: op> show ldp binding
Instance: default, AFI: ipv4, SAFI: labeled-unicast
  Prefix                                                In Label          Out Label        LDP ID             Status
  198.51.100.12/32                                             -                  3        -                  Best
  198.51.100.13/32                                             -                  3        -                  Best
  198.51.100.22/32                                             3                  -        198.51.100.21:0    Non-best
  198.51.100.23/32                                             3                  -        198.51.100.21:0    Non-best
Instance: default, AFI: ipv6, SAFI: labeled-unicast
  Prefix                                                In Label          Out Label        LDP ID             Status
  2001:db8:0:1::12/128                                         -                  3        -                  Best
  2001:db8:0:1::13/128                                         -                  3        -                  Best
  2001:db8:0:1::22/128                                         3                  -        198.51.100.21:0    Non-best
  2001:db8:0:1::23/128                                         3                  -        198.51.100.21:0    Non-best
Instance: default, AFI: l2vpn, SAFI: vpls-vpws
  Prefix                                                In Label          Out Label        LDP ID             Status
  00.00.00.00.00.00.00.00.00.00:100/112                        -       label:524032        -                  Best
                                                     label:524032                  -        198.51.100.21:0    Best
  00.00.00.00.00.00.00.00.00.00:200/112                        -       label:524033        -                  Best
                                                     label:524033                  -        198.51.100.21:0    Best
```

### show ldp route

**Syntax:**

**show ldp route** <options>

| Option | Description |
|---|---|
| - | Without any option, this command displays a summary of LDP route information. |
| instance <instance-name> | Displays LDP route information for the specified instance. |
| instance <instance-name> <afi> | Displays LDP route information for the specified address family and instance. Supported AFI values: ipv4, ipv6, mpls, l2vpn. |
| instance <instance-name> ipv4 prefix <ip> | Displays LDP route information for the specified address family of IPv4 prefix and instance. |
| instance <instance-name> ipv6 prefix <ip> | Displays LDP route information for the specified address family of IPv6 prefix and instance. |
| instance <instance-name> prefix <ip> | Displays LDP route information for the specified prefix and instance. |
| instance <instance-name> label <label> | Displays LDP route information for the specified MPLS label and instance. |
| instance <instance-name> mpls | Displays LDP route information of MPLS labels. |
| instance <instance-name> mpls label <label> | Displays LDP route information for the specified MPLS label and instance. |
| label <label> | Displays LDP route information for the specified MPLS label. |
| ipv4 | Displays LDP route information of the IPv4 address family. |
| ipv4 prefix <ip> | Displays LDP route IPv4 address family information for the specified prefix. |
| ipv6 | Displays LDP route information of the IPv6 address family. |
| ipv6 prefix <ip> | Displays LDP route IPv6 address family information for the specified prefix. |
| mpls | Displays LDP route information of MPLS labels. |
| mpls label <label> | Displays LDP route information for the specified MPLS label. |
| prefix <ip> | Displays LDP route information for the specified prefix address. |

Example: Summary view of the LDP route

```
supervisor@rtbrick: op> show ldp route
Instance: default, AFI: ipv4, SAFI: labeled-unicast
  Prefix               Out Label               Next Hop                                  Interface           Metric
  198.51.100.12/32     -                       -                                         -                   0
  198.51.100.13/32     -                       -                                         -                   0
Instance: default, AFI: ipv6, SAFI: labeled-unicast
  Prefix                                       Out Label               Next Hop
Interface           Metric
  2001:db8:0:1::12/128                         -                       -                                         -
0
  2001:db8:0:1::13/128                         -                       -                                         -
0
Instance: default, AFI: l2vpn, SAFI: vpls-vpws
  Prefix                                       Out Label               Next Hop
Interface           Metric
```

```
  00.00.00.00.00.00.00.00.00.00:100/112        524032                 198.51.100.21                            -
2000000
  00.00.00.00.00.00.00.00.00.00:200/112        524033                 198.51.100.21                            -
2000000
Instance: default, AFI: mpls, SAFI: unicast
  Label                Out Label              Next Hop                                  Interface           Metric
  524032               -                      -                                         ifl-0/0/5/100       -
  524032,bos:1         -                      -                                         ifl-0/0/5/100       -
  524033               -                      -                                         ifl-0/0/5/200       -
  524033,bos:1         -                      -                                         ifl-0/0/5/200       -
```

### show ldp l2vpn pseudowire

**Syntax:**

**show ldp l2vpn pseudowire** <option>

| Option | Description |
|---|---|
| `detail` | Displays detailed LDP L2VPN pseudowire information. |
| `instance <name>` | Displays pseudowire peer IP information for the specified instance. |
| `instance <name> detail` | Displays detailed pseudowire peer IP information for the specified instance. |
| instance <name> prefix <ip> | Displays pseudowire information for the specified instance and prefix. |
| instance <name> prefix <ip> detail | Displays detailed pseudowire information for the specified instance and prefix. |

Example 1: Display LDP L2VPN pseudowire information for instance `default`

```
supervisor@rtbrick: op> show ldp l2vpn pseudowire
Instance: default, AFI: l2vpn, SAFI: vpls-vpws
  Number of local interfaces: 2
    Peer IP                                     Interface Name       Prefix
Status              PW ID
    198.51.100.21                               ifl-0/0/5/100        00.00.00.00.00.00.00.00.00.00:100/112        Up
100
    198.51.100.21                               ifl-0/0/5/200        00.00.00.00.00.00.00.00.00.00:200/112        Up
200
```

### request switch-over

This command initiates switch tunnelling between primary and backup TLDP neighbors manually.

**Syntax:**

**request switch-over ldp l2vpn instance** <instance> **pseudowire** <pseudowire-id>

```
supervisor@rtbrick>SPINE01: cfg> request switch-over ldp l2vpn instance default
pseudowire 100
```

# 3.4. L2X

## 3.4.1. Layer 2 Cross-Connect

**Overview**

Layer 2 Cross-Connect (L2X) is used to directly connect two interfaces at Layer 2, allowing traffic to pass between them without layer 3 processing. It forwards Ethernet frames between two endpoints without MAC learning, switching, or IP routing. It provides isolated Ethernet connectivity for private line services or interconnecting customer sites.

L2X services provide Layer 2 connectivity between interfaces and support both local switching and remote transport use cases.

- Local L2X services switch traffic between interfaces on the same device.
- Remote L2X services extend Layer 2 connectivity across devices through an MPLS transport network.

L2X services can be configured as unidirectional or bidirectional, depending on the traffic forwarding requirements.

L2X supports:

- Building point-to-point Ethernet services.
- Extending Layer 2 connectivity between ports.
- Providing transparent VLAN transport.

L2X binds two circuits into a dedicated path that provides traffic isolation. Frames received on one endpoint are forwarded unchanged to the paired endpoint. For remote endpoints, frames are encapsulated (using MPLS) and transported across the provider network. At the egress endpoint, encapsulation is removed and the original frame is delivered. This ensures predictable forwarding.

You can configure L2X locally on a single device or remotely across devices using an underlying transport network.

**Types of L2X**

L2X has two deployment types:

**Local L2X**: It creates a direct Layer 2 link between two ports or VLANs on the same router. Data frames move between the interfaces, supporting either unidirectional or bidirectional traffic flow depending on configuration.

The following figure shows a local L2X scenario.

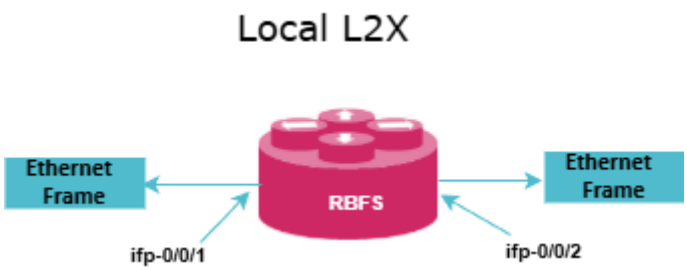



*Figure 51. Local L2X*

**Remote L2X**: It establishes a Layer 2 connection between ports on different devices. Each interface resides on a separate router, and an MPLS tunnel carries traffic between them. The two remote devices must be reachable over an MPLS path.

The following figure shows the remote L2X scenario.

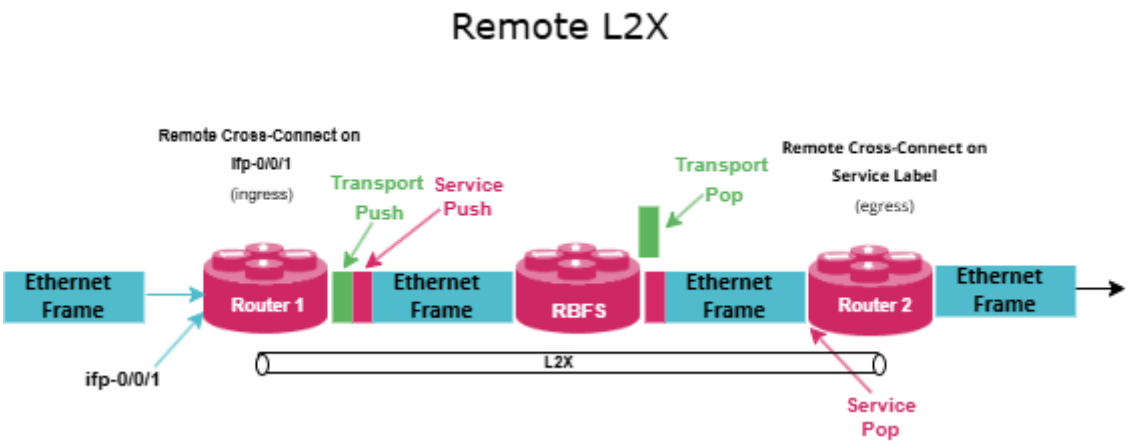



*Figure 52. Remote L2X*

#### Traffic Direction

L2X service direction (unidirectional and bidirectional) is independent of the L2X service type (local or remote). Unidirectional and bidirectional L2X define the traffic forwarding direction between endpoints.

A bidirectional forwarding behavior can be implemented either by using a single bidirectional L2X service or by configuring two complementary unidirectional L2X services, with one service handling each traffic direction independently.

**Unidirectional L2X**:

Unidirectional indicates data moves in just one direction. For example, traffic flows only from Interface A to Interface B.

**Bidirectional L2X**:

Traffic flows both directions. For example, traffic flows from Interface A to Interface B and from Interface B to Interface A. You can use L2X to cross-connect two local ports, enabling bidirectional traffic between them. This feature is supported only for local cross-connects.

VLAN manipulation is currently not supported on a single bidirectional local L2X

> service. However, this can be achieved by configuring two complementary unidirectional L2X services, with each service handling traffic in one direction independently.

#### Ingress and Egress Traffic in L2X

Ingress refers to traffic entering the network and egress refers to traffic leaving the network.

The ingress mode defines how incoming traffic is classified when it enters an L2X service endpoint. Traffic classification is performed based on the encapsulation type and VLAN information configured for the service.

#### Port and VLAN-based Cross-Connects

Port and VLAN-based cross-connects (L2X) are used to forward Layer 2 traffic between interfaces. RBFS supports both local and remote cross-connect deployments for port and VLAN modes.

Port refers to either a physical interface or a LAG interface, both of which are supported as L2X service endpoints.

In a port-based L2X service, all ingress traffic received on the configured port is forwarded through the cross-connect without VLAN-based classification. This includes untagged Ethernet frames, tagged traffic, and supported Layer 2 control packets received on the interface.

- Port-based cross-connects forward all traffic received on a port, regardless of VLAN tags. This mode operates purely at the port level and does not require VLAN configuration.
- VLAN-based cross-connects forward traffic selectively, based on a specific VLAN identifier.

#### VLAN Tagging Support

L2X supports multiple VLAN tagging scenarios:

- Single-tagged traffic
- Double-tagged traffic (Q-in-Q) with inner and outer VLAN tags
- Untagged traffic, supported only with port-based cross-connects

Double-tagged L2X services support traffic containing an inner VLAN tag encapsulated within an outer VLAN tag. The configuration is not limited to a specific VLAN ID combination and applies generally to double-tagged Ethernet traffic.

#### VLAN Operations Overview

RBFS supports VLAN manipulation on both ingress and egress interfaces. These operations allow modification of VLAN tags as traffic enters or leaves the device.

#### Supported VLAN Operations

The following table provides the supported VLAN operations:

| VLAN Operation | Description |
|---|---|
| `Single VLAN add` | Adds a VLAN tag (802.1Q or 802.1ad) to the frame. |
| `Single VLAN delete` | Removes a VLAN tag from the frame. |
| `Outer VLAN swap` | Replaces the outer VLAN tag with a new VLAN identifier. |

#### Encapsulation Support

L2X supports both 802.1Q and 802.1ad VLAN encapsulations, providing flexibility for deployments.

- Supports both the `802.1Q` and `802.1ad` encapsulations. Default encapsulation is `802.1Q`.
- Traffic is matched against both `802.1Q` and `802.1ad` Ethertypes by default in ingress mode.

#### Traffic Classification (Match Criteria)

Traffic classification determines how incoming traffic is matched before applying VLAN operations.

On a physical interface (`ifp`), traffic can be classified using the following match criteria:

| Match Criteria | Description |
|---|---|
| `(ifp)` | Matches all traffic on the interface (port-based). |
| `(ifp, outer_vlan)` | Matches traffic based on the outer VLAN tag. |
| `(ifp, outer_vlan, inner_vlan)` | Matches QinQ traffic with specific outer and inner VLAN tags. |
| `(ifp, outer_vlan, any inner_vlan)` | Matches traffic with a specific outer VLAN and any inner VLAN. |
| `(ifp, any vlan)` | Matches any tagged VLAN traffic on the interface. |

#### Configuration Rules and Restrictions

- Some match criteria are mutually exclusive. For example, `(ifp, outer_vlan, inner_vlan)` and `(ifp, outer_vlan, any inner_vlan)` cannot be configured together on the same interface.
- The `(ifp, any vlan)` match criteria cannot be combined with any other match type on the same interface.
- The `match-type` attribute must be explicitly configured for:
  - `match-untagged`
  - `match-any`
  - `match-inner-any`

#### VLAN Change Operation for IFLs

RBFS allows VLAN configurations changes on logical interfaces (IFLs) without removing the existing settings. This feature applies to both Layer 2 and Layer 3 IFLs.

**Supported Match Types**

The following table lists the supported match type options.

- The asterisk * indicates any VLAN or no VLAN tag.
- 'ov' indicates outer vlan and 'iv' indicates inner VLAN.

| Cases | Configuration A | Configuration B | Supported |
|---|---|---|---|
| **Case 1 : IFP A, *** | IFP A,* | IFP A, ov 10 | No |
| | IFP A,* | IFP A, ov 10, iv 20 | No |
| | IFP A,* | IFP A, ov 10, * | No |
| | IFP A,* | IFP A, untagged | No |
| **Case 2: IFP A, untagged** | IFP A, untagged | IFP A, * | No |
| | IFP A, untagged | IFP A, ov 10 | Yes |
| | IFP A, untagged | IFP A, ov 30, iv 20 | Yes |
| | IFP A, untagged | IFP A, ov 20, * | Yes |
| **Case 3: IFP A, outer_vlan:** | IFP A, ov 10 | IFP A, * | No |
| | IFP A, ov 10 | IFP A, ov 10, * | No |
| | IFP A, ov 10 | IFP A, ov 20 | Yes |
| | IFP A, ov 10 | IFP A, ov 10 , iv 20 | No |
| | IFP A, ov 10 | IFP A, ov 40 , iv 7 | Yes |
| | IFP A, ov 10 | IFP A, ov 30, * | Yes |
| | IFP A, ov 10 | IFP A, untagged | Yes |
| **Case 4: IFP A, outer_vlan, inner_vlan:** | IFP A, ov 10, iv 20 | IFP A, * | No |
| | IFP A, ov 10, iv 20 | IFP A, ov 10, * | No |
| | IFP A, ov 10, iv 20 | IFP A, ov 10 | No |
| | IFP A, ov 10, iv 20 | IFP A, ov 30 | Yes |
| | IFP A, ov 10, iv 20 | FP A, ov 20 , * | Yes |
| | IFP A, ov 10, iv 20 | IFP A, untagged | Yes |
| | IFP A, ov 10, iv 20 | IFP A, ov 10, iv 30 | Yes |

| Cases | Configuration A | Configuration B | Supported |
|---|---|---|---|
| **Case 5: IFP A, outer_vlan, *** | IFP A, ov 10, * | IFP A, * | No |
| | IFP A, ov 10, * | IFP A, ov 10 | No |
| | IFP A, ov 10, * | IFP A, ov 10, iv 20 | No |
| | IFP A, ov 10, * | IFP A, ov 20, iv 7 | Yes |
| | IFP A, ov 10, * | IFP A, ov 30 | Yes |
| | IFP A, ov 10, * | IFP A, untagged | Yes |
| | IFP A, ov 10, * | IFP A, ov 40, * | Yes |

#### Traffic Matching (Match Type)

A match type determines how incoming traffic is recognized and mapped to a particular Layer 2 cross-connect service. The match type specifies which attributes of an incoming packet are examined to associate it with the correct L2X instance.

The supported match type option includes:

| Match Type | Description |
|---|---|
| `match-any` | Matches all frames, regardless of VLAN tags. |
| `match-inner-any` | Matches frames based on the outer VLAN tag, irrespective of the inner VLAN tag. |
| `match-outer` | Matches frames based on the outer VLAN tag. |
| `match-outer-inner` | Matches frames based on both outer and inner VLAN tags. |
| `match-service-label` | Matches frames based on a specific MPLS service label. |
| `match-untagged` | Matches frames without any VLAN tags (untagged traffic). |

#### Information About IFL Traffic Statistics

IFL traffic statistics are counters collected on logical interfaces that track the volume of traffic flowing through the interface in ingress and egress directions. Statistics provide visibility into how traffic is handled at the interface level. By default, traffic statistics for Layer 2 IFLs and L2X are disabled. You must explicitly enable statistics using the `statistics true` command.

- Statistics can be enabled on up to 2048 interfaces, including:
  - Layer 2 IFLs
  - Layer 3 IFLs
  - L2X interfaces
- When QoS is enabled on a Layer 2 or Layer 3 IFL, the interface uses QoS counters and does not count toward the 2048-interface limit.
- After attaching or detaching a QoS profile, you must clear the IFL statistics to ensure accurate counter behavior.

- L2X supports statistics collection for both Ingress direction and Egress direction.
- For static L2X entries, statistics are available for only one direction. Ingress L2X supports ingress statistics only, while egress L2X supports egress statistics only.

#### Guidelines and Limitations

- **Bidirectional L2X:**
  - No VLAN operation is supported.
  - No statistics is supported.
- **Untagged traffic:**
  - Only port-based cross-connect is supported.
- **LLDP protocol packet cross-connects via L2X:**
  - By default, LLDP protocol packets are cross-connected in the RtBrick stack and are not punted to the control plane.

#### Supported Platforms

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

### 3.4.2. L2X Configuration

This section explains how to configure L2X services, detailing both creating local and remote cross-connect instances, defining endpoints, and specifying traffic-matching criteria.

#### Configuration Hierarchy

The diagram illustrates the L2X configuration hierarchy.

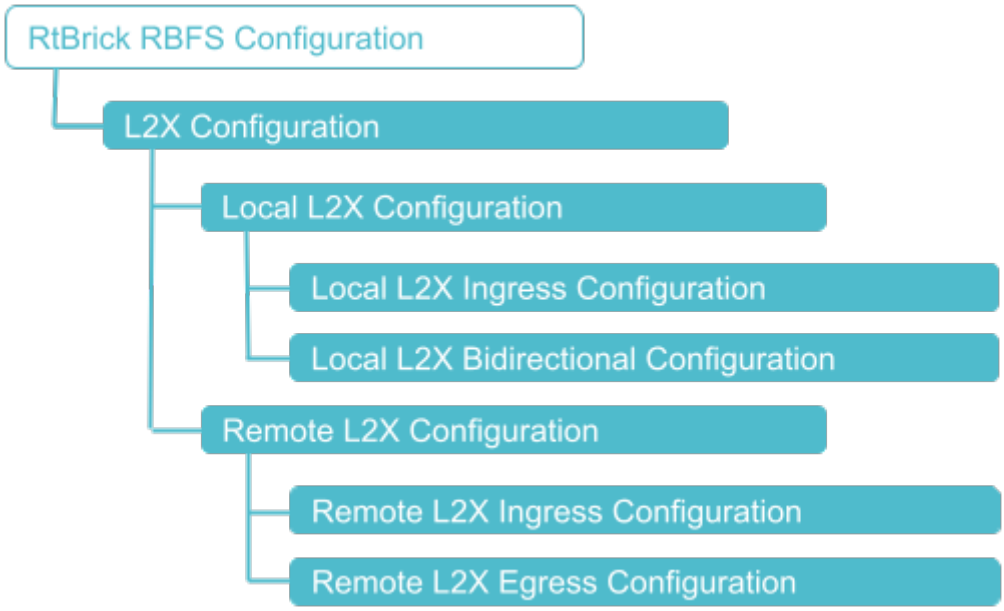



*Figure 53. L2X Configuration Hierarchy*

### Configuration Syntax and Commands

The following sections describe the L2X configuration syntax and commands.

#### Local L2X Configuration

The following sections describe the Local L2X unidirectional and bidirectional configurations.

##### Local L2X Ingress Configuration

This configuration enables local unidirectional L2X (Local Cross-Connect) on the same device.

**Syntax:**

**set l2x name** <name> **[ingress** | **bi-directional] description** <description>
**set l2x name** <name> **[ingress** | **bi-directional] egress-vlan-encapsulation [802.1q** | **802.1ad]**
**set l2x name** <name> **[ingress** | **bi-directional] egress-vlan-operation [single-vlan-add** | **single-vlan-delete** | **swap-outer-vlan]**
**set l2x name** <name> **[ingress** | **bi-directional] incoming-inner-vlan** <incoming-inner-vlan>
**set l2x name** <name> **[ingress** | **bi-directional] incoming-interface** <incoming-interface>
**set l2x name** <name> **[ingress** | **bi-directional] incoming-outer-vlan** <incoming-outer-vlan>
**set l2x name** <name> **[ingress** | **bi-directional] ingress-outer-vlan** <ingress-outer-vlan>
**set l2x name** <name> **[ingress** | **bi-directional] ingress-vlan-encapsulation [802.1q** | **802.1ad]**
**set l2x name** <name> **[ingress** | **bi-directional] ingress-vlan-operation [single-vlan-add** | **single-vlan-delete** | **swap-outer-vlan]**
**set l2x name** <name> **[ingress** | **bi-directional] match-tp-id [802.1q** | **802.1ad]**
**set l2x name** <name> **[ingress** | **bi-directional] match-type [match-any** | **match-untagged** | **match-inner-any** | **match-outer** | **match-outer-inner** | **match-service-label]**
**set l2x name** <name> **[ingress** | **bi-directional] outgoing-interface** <outgoing-interface>
**set l2x name** <name> **[ingress** | **bi-directional] outgoing-outer-vlan** <outgoing-outer-vlan>
**set l2x name** <name> **[ingress** | **bi-directional] statistics true**

| Attribute | Description |
|---|---|
| `<l2x-name>` | Name of L2X |
| `description <description>` | (Optional) L2X description |
| `egress-vlan-encapsulation <encapsulation>` | (Optional) Egress VLAN encapsulation value. NOTE: This option can be configured only after ‘swap-outer-vlan’ configuration. |
| `egress-vlan-operation <vlan-action>` | (Optional) Outgoing VLAN operation |
| `incoming-inner-vlan <vlan-id>` | (Optional) Incoming inner VLAN |
| `incoming-interface <incoming-interface>` | Incoming physical interface name |

| Attribute | Description |
|---|---|
| `incoming-outer-vlan <vlan-id>` | (Optional) Incoming outer VLAN |
| `ingress-outer-vlan <vlan-id>` | (Optional) Outer VLAN at ingress side |
| `ingress-vlan-encapsulation <encapsulation>` | (Optional) Ingress VLAN encapsulation value. NOTE: This option can be configured only after 'swap-outer-vlan' configuration. |
| `ingress-vlan-operation <vlan-action>` | (Optional) VLAN operation on ingress side outer VLAN |
| `match-type <match-type>` | L2X match type |
| `match-tp-id 802.1ad <tpid>` | The option enables L2X to recognize and handle packets tagged with 0x88a8/0x8100 TPID, allowing interoperability with 802.1ad/802.1q encapsulated traffic. |
| `outgoing-interface <outgoing-interface>` | Outgoing physical interface name |
| `outgoing-outer-vlan <vlan-id>` | (Optional) Outgoing outer VLAN |
| `statistics true` | Enables (true) L2X statistics. Default: Disabled.<br>For information on guidelines and limitation L2X statistics, see the Guidelines and Limitations section. |

Example: Local L2X Ingress Configuration with Port Match

The following configuration creates two local cross-connect (L2X) entries to forward traffic directly between the interfaces ifp-0/0/10 and ifp-0/0/11. One entry handles downstream traffic from ifp-0/0/10 to ifp-0/0/11, while the other handles upstream traffic in the reverse direction. The `match-any` option ensures that all ingress traffic on the specified interface is forwarded to the configured outgoing interface.

```
set l2x name local_cross_connect_ds ingress
set l2x name local_cross_connect_ds ingress incoming-interface ifp-0/0/10
set l2x name local_cross_connect_ds ingress outgoing-interface ifp-0/0/11
set l2x name local_cross_connect_ds ingress match-type match-any
set l2x name local_cross_connect_us ingress
set l2x name local_cross_connect_us ingress incoming-interface ifp-0/0/11
set l2x name local_cross_connect_us ingress outgoing-interface ifp-0/0/10
set l2x name local_cross_connect_us ingress match-type match-any
```

```
{
  "rtbrick-config:l2x": {
    "name": [
      {
        "name": "local_cross_connect_ds",
        "direction": "ingress",
        "incoming-interface": "ifp-0/0/10",
        "outgoing-interface": "ifp-0/0/11",
        "match-type": "match-any"
      },
```

```
      {
        "name": "local_cross_connect_us",
        "direction": "ingress",
        "incoming-interface": "ifp-0/0/11",
        "outgoing-interface": "ifp-0/0/10",
        "match-type": "match-any"
      }
    ]
  }
}
```

Example: Local L2X Ingress VLAN Operations

This configuration defines a local Layer 2 cross-connect for traffic arriving on 'ifp-0/0/10' with outer VLAN 200, forwarding it to ifp-0/0/11. The match condition is based on the outer VLAN (`match-outer`), ensuring only VLAN 200 traffic is processed. A VLAN operation is applied to add a single outer VLAN tag 400 to the traffic before forwarding.

```
set l2x name local_cross_connect_ds ingress
set l2x name local_cross_connect_ds ingress incoming-interface ifp-0/0/10
set l2x name local_cross_connect_ds ingress incoming-outer-vlan 200
set l2x name local_cross_connect_ds ingress outgoing-interface ifp-0/0/11
set l2x name local_cross_connect_ds ingress match-type match-outer
set l2x name local_cross_connect_ds ingress ingress-vlan-operation single-vlan-add
set l2x name local_cross_connect_ds ingress ingress-outer-vlan 400
```

```
{
  "rtbrick-config:l2x": {
    "name": [
      {
        "name": "local_cross_connect_ds",
        "direction": "ingress",
        "incoming-interface": "ifp-0/0/10",
        "incoming-outer-vlan": 200,
        "outgoing-interface": "ifp-0/0/11",
        "match-type": "match-outer",
        "ingress-vlan-operation": "single-vlan-add",
        "ingress-outer-vlan": 400
      }
    ]
  }
}
```

This configuration sets up a local Layer 2 cross-connect for traffic received on ifp-0/0/10 with outer VLAN 100, forwarding it to ifp-0/0/11 based on outer VLAN matching. On ingress, the outer VLAN tag is removed (`single-vlan-delete`), and on egress, the VLAN tag is re-added using a swap operation with VLAN 100 and '802.1ad' encapsulation. This configuration rewrites the VLAN tag

while maintaining service delimitation using the specified TPID.

```
set l2x name local_cross_connect_ds ingress incoming-interface ifp-0/0/10
set l2x name local_cross_connect_ds ingress incoming-outer-vlan 100
set l2x name local_cross_connect_ds ingress outgoing-interface ifp-0/0/11
set l2x name local_cross_connect_ds ingress egress-vlan-operation swap-outer-vlan
set l2x name local_cross_connect_ds ingress outgoing-outer-vlan 100
set l2x name local_cross_connect_ds ingress egress-vlan-encapsulation 802.1ad
set l2x name local_cross_connect_ds ingress match-type match-outer
set l2x name local_cross_connect_ds ingress ingress-vlan-operation single-vlan-delete
set l2x name local_cross_connect_ds ingress match-tp-id 802.1ad
```

```
{
  "rtbrick-config:l2x": {
    "name": [
      {
        "name": "local_cross_connect_ds",
        "direction": "ingress",
        "incoming-interface": "ifp-0/0/10",
        "incoming-outer-vlan": 100,
        "outgoing-interface": "ifp-0/0/11",
        "egress-vlan-operation": "swap-outer-vlan",
        "outgoing-outer-vlan": 100,
        "egress-vlan-encapsulation": "802.1ad",
        "match-type": "match-outer",
        "ingress-vlan-operation": "single-vlan-delete",
        "match-tp-id": "802.1ad"
      }
    ]
  }
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Local L2X Bidirectional Configuration

This configuration enables redirecting traffic incoming (ingress) on a particular interface to another interface and vice versa on the same hardware device.

**Syntax:**

**set l2x name** <name> **bi-directional match-type [match-any** | **match-untagged** | **match-inner-any** | **match-outer** | **match-outer-inner** | **match-service-label]**
**set l2x name** <name> **bi-directional incoming-interface** <incoming-interface>
**set l2x name** <name> **bi-directional outgoing-interface** <outgoing-interface>
**set l2x name** <name> **bi-directional description** <description>
**set l2x name** <name> **bi-directional egress-vlan-encapsulation [802.1q**| **802.1ad]**
**set l2x name** <name> **bi-directional incoming-inner-vlan** <incoming-inner-vlan>

**set l2x name** <name> **bi-directional incoming-outer-vlan** <incoming-outer-vlan>
**set l2x name** <name> **bi-directional ingress-outer-vlan** <ingress-outer-vlan>
**set l2x name** <name> **bi-directional ingress-vlan-encapsulation** <*802.1q* | **802.1ad**>
**set l2x name** <name> **bi-directional outgoing-outer-vlan** <outgoing-outer-vlan>
**set l2x name** <name> **bi-directional statistics true**

| Attribute | Description |
|---|---|
| `match-type <match-type>` | Match types with which traffic can be matched. |
| `incoming-interface <incoming-interface>` | Incoming interface is where the traffic originates from. |
| `outgoing-interface <outgoing-interface>` | Outgoing interface where the traffic is going to. |
| `description <description>` | (Optional) L2X description |
| `egress-vlan-encapsulation <encapsulation>` | (Optional) Egress VLAN encapsulation |
| `incoming-inner-vlan <vlan-id>` | (Optional) Incoming inner VLAN |
| `incoming-outer-vlan <vlan-id>` | (Optional) Incoming outer VLAN |
| `ingress-outer-vlan <vlan-id>` | (Optional) Outer VLAN at ingress side |
| `ingress-vlan-encapsulation <encapsulation>` | (Optional) Ingress VLAN encapsulation |
| `outgoing-outer-vlan <vlan-id>` | (Optional) Outgoing outer VLAN |
| `statistics true` | Enables bi-directional traffic statistics collection for this L2X connection. When set to true, the system collects and reports statistics for both directions of traffic. |

Example: Local L2X Bidirectional Configuration

This configuration creates a bidirectional local Layer 2 cross-connect between interfaces ifp-0/0/10 and ifp-0/0/11. It forwards traffic in both directions without requiring separate ingress entries for each path. The match-type 'match-untagged' ensures that only untagged traffic is matched and cross-connected between the interfaces.

```
set l2x name local_bidi bi-directional
set l2x name local_bidi bi-directional incoming-interface ifp-0/0/10
set l2x name local_bidi bi-directional outgoing-interface ifp-0/0/11
set l2x name local_bidi bi-directional match-type match-untagged
```

```
{
  "rtbrick-config:l2x": {
    "name": [
      {
        "name": "local_bidi",
        "direction": "bi-directional",
        "incoming-interface": "ifp-0/0/10",
```

```
        "outgoing-interface": "ifp-0/0/11",
        "match-type": "match-untagged"
      }
    ]
  }
}
```

#### Local L2X VLAN Operation Configuration

This configuration enables VLAN operations for both ingress and egress traffic on a particular interface to another interface and vice versa on the same hardware device.

**Syntax:**

**set l2x name** <name> **ingress [egress-vlan-operation | ingress-vlan-operation] [single-vlan-add | single-vlan-delete | swap-outer-vlan]**

| Attribute | Description |
|---|---|
| `<l2x-name>` | Name of the l2x. |
| `single-vlan-add` | Adds a single VLAN tag to the packet. |
| `single-vlan-delete` | Removes a single VLAN tag from the packet. |
| `swap-outer-vlan` | Replaces the outer VLAN tag with a new one. |

#### Remote L2X Configuration

The following sections describe the remote L2X configurations.

#### Remote L2X Ingress Configuration

This configuration enables the remote L2X ingress side.

**Syntax:**

**set l2x name** <name> **ingress match-type [match-any| match-untagged | match-inner-any| match-outer | match-outer-inner | match-service-label]**
**set l2x name** <name> **ingress incoming-interface** <incoming-interface>
**set l2x name** <name> **ingress ingress-vlan-operation [single-vlan-add | single-vlan-delete | swap-outer-vlan]**
**set l2x name** <name> **ingress ingress-outer-vlan** <ingress-outer-vlan>
**set l2x name** <name> **ingress ingress-vlan-encapsulation [802.1q | 802.1ad]**
**set l2x name** <name> **ingress nexthop4** <nexthop4>
**set l2x name** <name> **ingress nexthop6** <nexthop6>
**set l2x name** <name> **ingress lookup-instance** <lookup-instance>
**set l2x name** <name> **ingress lookup-afi [ipv4 | ipv6 | mpls]**
**set l2x name** <name> **ingress lookup-safi [labeled-unicast | multicast | unicast]**
**set l2x name** <name> **ingress service-label** <service-label>

**set l2x name** <name> **ingress statistics true**

| Attribute | Description |
|---|---|
| `match-type <match-type>` | Match types with which traffic can be matched. The 'match-type' option includes `match-any`, `match-inner-any`, `match-outer`, `match-outer-inner`, `match-service-label`, and `match-untagged`. For more details, see the Match Type Options section. |
| `incoming-interface <incoming-interface>` | Incoming interface is where the traffic originates from. |
| `ingress-vlan-operation <ingress-vlan-action>` | (Optional) VLAN operation on ingress side outer VLAN |
| `ingress-outer-vlan <vlan-id>` | (Optional) Outer VLAN at ingress side |
| `ingress-vlan-encapsulation <encapsulation>` | (Optional) Ingress VLAN encapsulation value |
| `nexthop4/nexthop6 <nexthop>` | Next-Hop address |
| `lookup-instance <lookup-instance>` | (Optional) Instance name |
| `lookup-afi <lookup-afi>` | (Optional) AFI value: IPv4, IPv6, and MPLS. |
| `lookup-safi <lookup-safi>` | (Optional) SAFI value: safi values are unicast, labeled-unicast, and multicast. |
| `service-label <service_label>` | Service label value.<br>For details on the supported label ranges, see the Static, System, and Dynamic Label Ranges section in the Platform Guide. |
| `statistics true` | Enables bi-directional traffic statistics collection for this L2X connection. When set to true, the system collects and reports statistics for both directions of traffic. |

Example: Remote L2X Ingress Configuration with Port Match

This configuration defines a remote Layer 2 cross-connect for traffic arriving on ifp-0/0/10, forwarding it towards a remote endpoint using the next-hop 198.51.100.44. The forwarding decision is performed via the default routing instance using IPv4 labeled-unicast lookup, with service label 10000 applied for MPLS-based transport. The match-type 'match-any' ensures that all incoming traffic on the interface is processed by this cross-connect.

```
set l2x name remote_l2x ingress incoming-interface ifp-0/0/10
set l2x name remote_l2x ingress nexthop4 198.51.100.44
set l2x name remote_l2x ingress lookup-instance default
set l2x name remote_l2x ingress lookup-afi ipv4
set l2x name remote_l2x ingress lookup-safi labeled-unicast
set l2x name remote_l2x ingress service-label 10000
```

```
set l2x name remote_l2x ingress match-type match-any
```

```
{
  "rtbrick-config:l2x": {
    "name": [
      {
        "name": "remote_l2x",
        "direction": "ingress",
        "incoming-interface": "ifp-0/0/10",
        "nexthop4": "198.51.100.44",
        "lookup-instance": "default",
        "lookup-afi": "ipv4",
        "lookup-safi": "labeled-unicast",
        "service-label": 10000,
        "match-type": "match-any"
      }
    ]
  }
}
```

Example: Remote L2X Ingress Configuration with VLAN match

This configuration defines a remote Layer 2 cross-connect for double-tagged traffic received on ifp-0/0/10 with outer VLAN 100 and inner VLAN 200. The traffic is matched using match-outer-inner and forwarded to a remote endpoint via next-hop 198.51.100.44 using IPv4 labeled-unicast lookup in the default instance. A service label 10000 is applied to enable MPLS-based transport of the matched traffic.

```
set l2x name remote_l2x ingress incoming-interface ifp-0/0/10
set l2x name remote_l2x ingress incoming-outer-vlan 100
set l2x name remote_l2x ingress incoming-inner-vlan 200
set l2x name remote_l2x ingress nexthop4 198.51.100.44
set l2x name remote_l2x ingress lookup-instance default
set l2x name remote_l2x ingress lookup-afi ipv4
set l2x name remote_l2x ingress lookup-safi labeled-unicast
set l2x name remote_l2x ingress service-label 10000
set l2x name remote_l2x ingress match-type match-outer-inner
```

```
{
  "rtbrick-config:l2x": {
    "name": [
      {
        "name": "remote_l2x",
        "direction": "ingress",
        "incoming-interface": "ifp-0/0/10",
        "incoming-outer-vlan": 100,
        "incoming-inner-vlan": 200,
```

```
        "nexthop4": "198.51.100.44",
        "lookup-instance": "default",
        "lookup-afi": "ipv4",
        "lookup-safi": "labeled-unicast",
        "service-label": 10000,
        "match-type": "match-outer-inner"
      }
    ]
  }
}
```

Example: Remote L2X Ingress VLAN operations

This configuration defines a remote Layer 2 cross-connect for double-tagged traffic received on ifp-0/0/10 with outer VLAN 100 and inner VLAN 200, using 'match-outer-inner' for classification. The matched traffic has one VLAN tag removed on ingress (`single-vlan-delete`) before being forwarded to the remote endpoint via next-hop 198.51.100.44 using IPv4 labeled-unicast lookup. A service label 10000 is applied to carry the traffic over MPLS.

```
set l2x name remote_l2x ingress incoming-interface ifp-0/0/10
set l2x name remote_l2x ingress incoming-outer-vlan 100
set l2x name remote_l2x ingress incoming-inner-vlan 200
set l2x name remote_l2x ingress nexthop4 198.51.100.44
set l2x name remote_l2x ingress lookup-instance default
set l2x name remote_l2x ingress lookup-afi ipv4
set l2x name remote_l2x ingress lookup-safi labeled-unicast
set l2x name remote_l2x ingress service-label 10000
set l2x name remote_l2x ingress match-type match-outer-inner
set l2x name remote_l2x ingress ingress-vlan-operation single-vlan-delete
```

```
{
  "rtbrick-config:l2x": {
    "name": [
      {
        "name": "remote_l2x",
        "direction": "ingress",
        "incoming-interface": "ifp-0/0/10",
        "incoming-outer-vlan": 100,
        "incoming-inner-vlan": 200,
        "nexthop4": "198.51.100.44",
        "lookup-instance": "default",
        "lookup-afi": "ipv4",
        "lookup-safi": "labeled-unicast",
        "service-label": 10000,
        "match-type": "match-outer-inner",
        "ingress-vlan-operation": "single-vlan-delete"
      }
    ]
```

```
  }
}
```

#### Remote L2X Egress Configuration

This configuration enables the remote L2X egress side.

**Syntax:**

**set l2x name** <name> **egress outgoing-interface** <outgoing-interface>
**set l2x name** <name> **egress egress-vlan-operation [single-vlan-add** | **single-vlan-delete** | **swap-outer-vlan]**
**set l2x name** <name> **egress egress-outer-vlan** <egress-outer-vlan>
**set l2x name** <name> **egress egress-vlan-encapsulation [802.1q** | **802.1ad]**
**set l2x name** <name> **egress service-label** <service-label>
**set l2x name** <name> **egress statistics true**

| Attribute | Description |
|---|---|
| `match-type <match-type>` | Match types with which traffic can be matched. The 'match-type' option includes `match-any`, `match-inner-any`, `match-outer`, `match-outer-inner`, `match-service-label`, and `match-untagged`. For more details, see the Match Type Options section. |
| `outgoing-interface <outgoing-interface>` | Interface where traffic is going to. |
| `service-label <service_label>` | Service label value.<br>For details on the supported label ranges, see the Static, System, and Dynamic Label Ranges section in the Platform Guide. |
| `nexthop4/nexthop6 <nexthop>` | Next-Hop address |
| `lookup-instance <lookup-instance>` | (Optional) Instance name |
| `lookup-afi <lookup-afi>` | (Optional) AFI value: IPv4, IPv6, and MPLS. |
| `lookup-safi <lookup-safi>` | (Optional) SAFI value: safi values are unicast, labeled-unicast, and multicast. |
| `egress-vlan-operation <vlan-action>` | (Optional) Outgoing VLAN operation. |
| `outgoing-outer-vlan <vlan-id>` | (Optional) Outgoing outer VLAN. |
| `statistics true` | Enables bi-directional traffic statistics collection for this L2X connection. When set to true, the system collects and reports statistics for both directions of traffic. |

Example: Remote L2X Egress Configuration

This configuration defines the egress handling for a remote Layer 2 cross-connect with service label 10000. Traffic received over MPLS with the matching service label is decapsulated and forwarded to the local interface ifp-0/0/11. This enables proper termination of the remote L2X service on the device.

```
set l2x name remote_l2x egress
set l2x name remote_l2x egress service-label 10000
set l2x name remote_l2x egress outgoing-interface ifp-0/0/11
```

```
{
  "rtbrick-config:l2x": {
    "name": [
      {
        "name": "remote_l2x",
        "direction": "egress",
        "service-label": 10000,
        "outgoing-interface": "ifp-0/0/11"
      }
    ]
  }
}
```

Example: Local L2X Egress Configuration with VLAN Operation

This configuration defines the egress handling for a remote Layer 2 cross-connect with service label 10000, forwarding traffic to ifp-0/0/11. Upon egress, a VLAN tag 400 is added to the traffic using `single-vlan-add`. This ensures the decapsulated MPLS traffic is properly VLAN-tagged before being transmitted on the outgoing interface.

```
set l2x name remote_l2x egress service-label 10000
set l2x name remote_l2x egress outgoing-interface ifp-0/0/11
set l2x name remote_l2x egress egress-vlan-operation single-vlan-add
set l2x name remote_l2x egress outgoing-outer-vlan 400
```

```
{
  "rtbrick-config:l2x": {
    "name": [
      {
        "name": "remote_l2x",
        "direction": "egress",
        "service-label": 10000,
        "outgoing-interface": "ifp-0/0/11",
        "egress-vlan-operation": "single-vlan-add",
        "outgoing-outer-vlan": 400
      }
    ]
```

```
  }
}
```

#### L2X over LAG Interfaces

RBFS L2X services to operate over LAG interfaces for remote attachment circuits.

Example:

The following configuration enables an L2X service named 'lag1' to forward traffic between LAG interfaces based on outer VLAN matching. In the bi-directional configuration, traffic received on interface lag-1 with outer VLAN 100 is forwarded to lag-2. The ingress configuration defines one-way forwarding from lag-1 (outer VLAN 100) to lag-2. In both cases, the match-type `match-outer `setting ensures that forwarding decisions are based on the outer VLAN tag. This configuration enables VLAN-based Layer 2 traffic forwarding between lag-1 and lag-2.

```
set l2x name lag1 bi-directional incoming-interface lag-1
set l2x name lag1 bi-directional incoming-outer-vlan 100
set l2x name lag1 bi-directional outgoing-interface lag-2
set l2x name lag1 bi-directional match-type match-outer
set l2x name lag1 ingress incoming-interface lag-1
set l2x name lag1 ingress incoming-outer-vlan 100
set l2x name lag1 ingress outgoing-interface lag-2
set l2x name lag1 ingress match-type match-outer
```

```
supervisor@rtbrick.net: cfg> show config l2x
{
  "rtbrick-config:l2x": {
    "name": [
      {
        "name": "lag1",
        "direction": "bi-directional",
        "incoming-interface": "lag-1",
        "incoming-outer-vlan": 100,
        "outgoing-interface": "lag-2",
        "match-type": "match-outer"
      },
      {
        "name": "lag1",
        "direction": "ingress",
        "incoming-interface": "lag-1",
        "incoming-outer-vlan": 100,
        "outgoing-interface": "lag-2",
        "match-type": "match-outer"
      }
    ]
  }
}
```

### 3.4.3. L2X Operational Commands

The L2X show commands provide detailed information about the L2X operations.

#### L2X Show Commands

The L2X show commands display data from FIB local table. Therefore, local L2X with down ports or remote l2x with unresolved nexthop address are not displayed.

##### L2X Details

The show commands display L2X information, including total entry count, entry details, filtered L2X output, and L2X traffic statistics.

**Syntax:**

**show l2x**
**show l2x count**
**show l2x filter** <option>
**show l2x detail**
**show l2x statistics**

| Option | Description |
|---|---|
| `<cr>` | Without any option, the command displays all L2X information such as ingress L2X and egress L2X. |
| `count` | Displays the total number of L2X entries. |
| `detail` | Displays detailed information for L2X entries. |
| `filter <option>` | Displays L2X information based on specified filter criteria. |
| `statistics` | Displays L2X traffic statistics. |

Example: Summary of total L2X entries

The `show l2x` command without any option displays a summary of L2X entries.

```
supervisor@rtbrick: op> show l2x
Name               Direction       Incoming Intf        Status  Outer VLAN  Inner VLAN   Service label   Outgoing Intf
Status  NextHop
IFP-CE1-PE1        ingress         lag-1                up      501         -            -               ifp-0/1/20
up      -
IFP-PE1-CE1-501    ingress         ifp-0/1/20           up      501         -            -               lag-1
up      -
l2bsa_lag_1_101    ingress         lag-1                up      101         -            10001           ifl-0/1/20/0
up      192.0.2.2
l2bsa_lag_1_101    egress          -                    -       -           -            20001           lag-1
up      -
l2bsa_lag_1_111    ingress         lag-1                up      111         -            10011           ifl-0/1/20/0
up      192.0.2.2
l2bsa_lag_1_111    egress          -                    -       -           -            20011           lag-1
up      -
l2bsa_lag_1_121    ingress         lag-1                up      121         -            10021           ifl-0/1/20/0
```

```
up      192.0.2.2
l2bsa_lag_1_121    egress          -                   -      -          -            20021           lag-1
up      -
l2bsa_lag_1_131    ingress         lag-1               up     131        -            10031           ifl-0/1/20/0
up      192.0.2.2
l2bsa_lag_1_131    egress          -                   -      -          -            20031           lag-1
up      -
l2bsa_lag_1_141    ingress         lag-1               up     141        -            10041           ifl-0/1/20/0
up      192.0.2.2
l2bsa_lag_1_141    egress          -                   -      -          -            20041           lag-1
up      -
l2bsa_lag_1_201    ingress         lag-1               up     201        -            10201           ifl-0/1/20/0
up      2001:db8:1:1::2
l2bsa_lag_1_201    egress          -                   -      -          -            20201           lag-1
up      -
l2bsa_lag_1_211    ingress         lag-1               up     211        -            10211           ifl-0/1/20/0
up      2001:db8:1:1::2
l2bsa_lag_1_211    egress          -                   -      -          -            20211           lag-1
up      -
l2bsa_lag_1_221    ingress         lag-1               up     221        -            10221           ifl-0/1/20/0
up      2001:db8:1:1::2
l2bsa_lag_1_221    egress          -                   -      -          -            20221           lag-1
up      -
l2bsa_lag_1_231    ingress         lag-1               up     231        -            10231           ifl-0/1/20/0
up      2001:db8:1:1::2
l2bsa_lag_1_231    egress          -                   -      -          -            20231           lag-1
up      -
l2bsa_lag_1_241    ingress         lag-1               up     241        -            10241           ifl-0/1/20/0
up      2001:db8:1:1::2
l2bsa_lag_1_241    egress          -                   -      -          -            20241           lag-1
up      -
```

Example: L2X filter direction

The `show l2x filter direction ingress type remote` output lists all ingress remote L2X entries, showing their VLAN/service-label mapping, incoming and outgoing interfaces, operational status, and the remote next-hop address.

```
supervisor@rtbrick: op> show l2x filter direction ingress
Name               Direction       Incoming Intf        Status  Outer VLAN  Inner VLAN   Service label   Outgoing Intf
Status  NextHop
IFP-CE1-PE1        ingress         lag-1               up     501        -            -               ifp-0/1/20
up      -
IFP-PE1-CE1-501    ingress         ifp-0/1/20          up     501        -            -               lag-1
up      -
l2bsa_lag_1_101    ingress         lag-1               up     101        -            10001           ifl-0/1/20/0
up      192.0.2.2
l2bsa_lag_1_111    ingress         lag-1               up     111        -            10011           ifl-0/1/20/0
up      192.0.2.2
l2bsa_lag_1_121    ingress         lag-1               up     121        -            10021           ifl-0/1/20/0
up      192.0.2.2
l2bsa_lag_1_131    ingress         lag-1               up     131        -            10031           ifl-0/1/20/0
up      192.0.2.2
l2bsa_lag_1_141    ingress         lag-1               up     141        -            10041           ifl-0/1/20/0
up      192.0.2.2
l2bsa_lag_1_201    ingress         lag-1               up     201        -            10201           ifl-0/1/20/0
up      2001:db8:1:1::2
l2bsa_lag_1_211    ingress         lag-1               up     211        -            10211           ifl-0/1/20/0
up      2001:db8:1:1::2
l2bsa_lag_1_221    ingress         lag-1               up     221        -            10221           ifl-0/1/20/0
up      2001:db8:1:1::2
l2bsa_lag_1_231    ingress         lag-1               up     231        -            10231           ifl-0/1/20/0
up      2001:db8:1:1::2
```

```
l2bsa_lag_1_241    ingress         lag-1               up      241        -           10241          ifl-0/1/20/0
up      2001:db8:1:1::2
```

Example for detailed view of L2X information.

The `show l2x detail` command displays detailed information for each L2X entry, including direction, download status, VLAN match criteria, interface status and so on.

```
supervisor@rtbrick: op> show l2x detail | head -n 15
L2X name: IFP-CE1-PE1
  Direction: ingress
  Status: Download success
  Incoming interface: lag-1
  Incoming interface status: up
  Subtype: Incoming Port - Outer Vlan Match
  Incoming outer VLAN: 501
  Outgoing interface: ifp-0/1/20
  Outgoing interface status: up
  NextHop:
    NextHop type: Local egress cross connect
    NextHop action: No vlan manipulation - l2 forward
L2X name: IFP-PE1-CE1-501
  Direction: ingress
  Status: Download success
```

#### L2X Filter

The command displays L2X entries based on filter criteria such as interface, VLAN, next-hop, direction, service label, and so on.

**Syntax:**

**show l2x filter count**
**show l2x filter detail**
**show l2x filter direction [ingress** | **egress** | **bi-directional]**
**show l2x filter incoming-interface** <incoming-interface-name>
**show l2x filter inner-vlan** [1 - 65535]
**show l2x filter name** <l2x-name>
**show l2x filter nexthop4** <nexthop4>
**show l2x filter nexthop6** <nexthop6>
**show l2x filter outer-vlan** [1 - 65535]
**show l2x filter outgoing-interface** <outgoing-interface-name>
**show l2x filter service-label** <service-label>
**show l2x filter type [local** | **remote]**

| Option | Description |
|---|---|
| `<cr>` | Without any option, displays the L2X filter output using the configured filter criteria. |
| `count` | Filters the output by the total L2X entry count. |
| `detail` | Displays detailed information for matching L2X entries. |
| `direction` | Filters L2X entries by traffic direction. |
| `incoming-interface` | Filters L2X entries by the incoming interface name. |
| `inner-vlan` | Filters L2X entries by the inner VLAN ID. |
| `<l2x_name>` | Filters the output by a specific L2X name. |
| `nexthop4` | Filters L2X entries by the remote IPv4 next-hop address. |
| `nexthop6` | Filters L2X entries by the remote IPv6 next-hop address. |
| `outer-vlan` | Filters L2X entries by the outer VLAN ID. |
| `outgoing-interface` | Filters L2X entries by the outgoing interface name. |
| `service-label` | Filters L2X entries by the service label value. |
| `type <local \| remote>` | Filters L2X entries by the L2X type. |

API To access the Operational State API that corresponds to this CLI, click here.

Example: Output of L2X Statistics command

The `show l2x statistics` command output displays ingress and egress traffic counters for the specified L2X, including forwarded and dropped packets and bytes.

```
supervisor@rtbrick.net: cfg> show l2x statistics l2bsa_lag_1_111
L2X Name: l2bsa_lag_1_111
  Ingress
    Counters:
        In-Forward-Packets: 0
        In-Forward-Bytes: 0
        In-Drop-Packets: 0
        In-Drop-Bytes: 0
        Out-Forward-Packets: 0
        Out-Forward-Bytes: 0
        Out-Drop-Packets: 0
        Out-Drop-Bytes: 0
L2X Name: l2bsa_lag_1_111
  Egress
    Counters:
        In-Forward-Packets: 0
        In-Forward-Bytes: 0
        In-Drop-Packets: 0
        In-Drop-Bytes: 0
        Out-Forward-Packets: 0
        Out-Forward-Bytes: 0
```

```
        Out-Drop-Packets: 0
        Out-Drop-Bytes: 0
```

Example for the specified service label.

The `show l2x filter service-label <label>` command displays L2X filter entries matching the specified service label, VLAN match criteria, and interface status.

```
supervisor@rtbrick: op> show l2x filter service-label 10221
Name               Direction       Incoming Intf        Status  Outer VLAN  Inner VLAN   Service label   Outgoing Intf
Status  NextHop
l2bsa_lag_1_221    ingress         lag-1                up      221         -            10221           ifl-0/1/20/0
up      2001:db8:1:1::2
```

Example for the specified L2X filter type

The `show l2x filter type remote` command displays the remote L2X filter entries for ingress and egress traffic, including VLAN match criteria, service labels, and interface status.

```
supervisor@rtbrick: op> show l2x filter type remote
Name               Direction       Incoming Intf        Status  Outer VLAN  Inner VLAN   Service label   Outgoing Intf
Status  NextHop
l2bsa_lag_1_101    ingress         lag-1                up      101         -            10001           ifl-0/1/20/0
up      192.0.2.2
l2bsa_lag_1_101    egress          -                    -       -           -            20001           lag-1
up      -
l2bsa_lag_1_111    ingress         lag-1                up      111         -            10011           ifl-0/1/20/0
up      192.0.2.2
l2bsa_lag_1_111    egress          -                    -       -           -            20011           lag-1
up      -
l2bsa_lag_1_121    ingress         lag-1                up      121         -            10021           ifl-0/1/20/0
up      192.0.2.2
l2bsa_lag_1_121    egress          -                    -       -           -            20021           lag-1
up      -
l2bsa_lag_1_131    ingress         lag-1                up      131         -            10031           ifl-0/1/20/0
up      192.0.2.2
l2bsa_lag_1_131    egress          -                    -       -           -            20031           lag-1
up      -
l2bsa_lag_1_141    ingress         lag-1                up      141         -            10041           ifl-0/1/20/0
up      192.0.2.2
l2bsa_lag_1_141    egress          -                    -       -           -            20041           lag-1
up      -
l2bsa_lag_1_201    ingress         lag-1                up      201         -            10201           ifl-0/1/20/0
up      2001:db8:1:1::2
l2bsa_lag_1_201    egress          -                    -       -           -            20201           lag-1
up      -
l2bsa_lag_1_211    ingress         lag-1                up      211         -            10211           ifl-0/1/20/0
up      2001:db8:1:1::2
l2bsa_lag_1_211    egress          -                    -       -           -            20211           lag-1
up      -
l2bsa_lag_1_221    ingress         lag-1                up      221         -            10221           ifl-0/1/20/0
up      2001:db8:1:1::2
l2bsa_lag_1_221    egress          -                    -       -           -            20221           lag-1
up      -
l2bsa_lag_1_231    ingress         lag-1                up      231         -            10231           ifl-0/1/20/0
up      2001:db8:1:1::2
l2bsa_lag_1_231    egress          -                    -       -           -            20231           lag-1
up      -
l2bsa_lag_1_241    ingress         lag-1                up      241         -            10241           ifl-0/1/20/0
```

```
up      2001:db8:1:1::2
l2bsa_lag_1_241    egress          -                   -       -           -            20241           lag-1
up      -
```

#### L2X Hierarchy

The `show l2x hierarchy` command displays L2X information in a hierarchical format, showing each instance's direction-specific configuration and operational state across both control and forwarding planes.

**Syntax:**

**show l2x hierarchy** <l2x_name> **direction [ingress** | **egress** | **bi-directional]** <option>

| Option | Description |
|---|---|
| `<l2x-name>` | Displays the hierarchy for a specific user-defined L2X instance name. |
| `ingress` | Displays hierarchies for the traffic entering the router. |
| `egress` | Displays hierarchies for the outgoing traffic. |
| `bi-directional` | Displays both the Ingress and Egress hierarchies simultaneously. |
| `detail` | Provides the detailed information. |

The following example output shows the ingress L2X configuration for local L2X. The information is structured in a hierarchical order for each daemon. It matches traffic on interface 'ifp-0/1/51' with outer VLAN 200 and adding outer VLAN 400. The 'staticd' and 'ribd' sections display the forwarding parameters. The 'fibd' section confirms that the entry is successfully programmed.

```
supervisor@rtbrick.net: op> show l2x hierarchy l2x_local direction ingress
staticd:
  l2x name: l2x_local
  direction: ingress
  incoming ifp: ifp-0/1/51
  incoming outer vlan: 200
  vlan operation: single-vlan-add
  outer vlan: 400
    nexthop set hash: 88815a358f521438074f8207ea9196c5b17116cbf16ee70f
    nexthop hash: a70e95a94e65f1583a97e086c2c6df72f95e59ed07d981d8
      outgoing ifp: ifp-0/1/52
ribd:
  l2x name: l2x_local
  direction: ingress
  incoming interface: ifp-0/1/51
  incoming outer vlan: 200
  vlan operation: single-vlan-add
  outer vlan: 400
    adjacency set hash: 88815a358f521438074f8207ea9196c5b17116cbf16ee70f
```

```
    nexthop hash: a70e95a94e65f1583a97e086c2c6df72f95e59ed07d981d8
      outgoing ifp: ifp-0/1/52
fibd:
  l2x name: l2x_local
  direction: ingress
  incoming interface: ifp-0/1/51
  incoming outer vlan: 200
  vlan operation: single-vlan-add
  outer vlan: 400
    bcm status: Download success
    adjacency set hash: 88815a358f521438074f8207ea9196c5b17116cbf16ee70f
    nexthop hash: a70e95a94e65f1583a97e086c2c6df72f95e59ed07d981d8
      outgoing ifp: ifp-0/1/52
      adjacency status: Success
```

Example for remote L2X hierarchy information

This output displays the hierarchical ingress L2X information for the L2X remote instance. The details include configuration, routing adjacency, and hardware programming state. The 'staticd' and 'ribd' sections show VLAN delete operations and service label association. The 'fibd' section shows that the corresponding adjacencies and service label are successfully programmed in hardware.

```
supervisor@rtbrick.net: op> show l2x hierarchy l2x_remote direction ingress
staticd:
  l2x name: l2x_remote
  direction: egress
    nexthop set hash: b4097df5d61079a99bafd02ed2ab12d9125c02e9e0688b45
    vlan operation: single-vlan-delete
ribd:
  l2x name: l2x_remote
  direction: ingress
  incoming interface: ifp-0/1/51
  incoming outer vlan: 100
  incoming inner vlan: 200
  vlan operation: single-vlan-delete
  service label: 10000
    adjacency set hash: daf65a1cf024de74eb406450cdb4e8e1a163724811158c4e
    nexthop hash: 02c86c766b675e4a693b78ef395a65a1a1c6793e77629142
      nexthop: 100.1.1.1, exit interface: ifl-0/1/51/10
    adjacency set hash: daf65a1cf024de74eb406450cdb4e8e1a163724811158c4e
    nexthop hash: 02c86c766b675e4a693b78ef395a65a1a1c6793e77629142
      nexthop: 100.1.1.1, exit interface: ifl-0/1/52/10
fibd:
  l2x name: l2x_remote
  direction: egress
    bcm status: Download success
    adjacency set hash: daf65a1cf024de74eb406450cdb4e8e1a163724811158c4e
    adjacency status: Success
    bcm status: Download success
```

```
    adjacency set hash: daf65a1cf024de74eb406450cdb4e8e1a163724811158c4e
    adjacency status: Success
    service label: label:10000
```

Example for detailed view of remote L2X hierarchy.

This following example output provides a detailed hierarchical view of the L2X remote ingress instance with in-depth next-hop and lookup information. It shows MPLS label push actions, resolved next-hop attributes, and lookup context details from the RIB.

```
supervisor@rtbrick.net: op> show l2x hierarchy l2x_remote direction ingress detail
staticd:
  l2x name: l2x_remote
  direction: egress
    nexthop set hash: b4097df5d61079a99bafd02ed2ab12d9125c02e9e0688b45
    vlan operation: single-vlan-delete
ribd:
  l2x name: l2x_remote
  direction: ingress
  incoming interface: ifp-0/1/51
  incoming outer vlan: 100
  incoming inner vlan: 200
  vlan operation: single-vlan-delete
  service label: 10000
    adjacency set hash: daf65a1cf024de74eb406450cdb4e8e1a163724811158c4e
    nexthop hash: 02c86c766b675e4a693b78ef395a65a1a1c6793e77629142
      nexthop: 100.1.1.1, exit interface: ifl-0/1/51/10
      nexthop type: Remote ingress cross connect, nexthop action: mpls label push
      lookup instance: default, lookup afi: ipv4, lookup safi: unicast
      mac address: 7a:21:db:c0:00:05
      covering prefix: 100.1.1.1/32
      expanded nexthop: 10.1.1.2
      lookup instance id: 1
    adjacency set hash: daf65a1cf024de74eb406450cdb4e8e1a163724811158c4e
    nexthop hash: 02c86c766b675e4a693b78ef395a65a1a1c6793e77629142
      nexthop: 100.1.1.1, exit interface: ifl-0/1/52/10
      nexthop type: Remote ingress cross connect, nexthop action: mpls label push
      lookup instance: default, lookup afi: ipv4, lookup safi: unicast
      mac address: 7a:21:db:c0:00:06
      covering prefix: 100.1.1.1/32
      expanded nexthop: 20.1.1.2
      lookup instance id: 1
fibd:
  l2x name: l2x_remote
  direction: egress
    bcm status: Download success
    bcm type: MPLS port to Vlan port
    adjacency set hash: daf65a1cf024de74eb406450cdb4e8e1a163724811158c4e
    type: Remote ingress cross connect, action: mpls label push
    adjacency status: Success
```

```
    bcm status: Download success
    bcm type: MPLS port to Vlan port
    adjacency set hash: daf65a1cf024de74eb406450cdb4e8e1a163724811158c4e
    type: Remote ingress cross connect, action: mpls label push
    adjacency status: Success
    service label: label:10000
```

### L2X Clear Commands

Clear commands allow to reset operational states.

#### L2X Statistics

This commands resets L2X statistics for a specified Layer 2 cross connect instance.

**Syntax:**

**clear l2x statistics** <l2x-name>

| Attribute | Description |
|---|---|
| `<l2x-name>` | L2X name. |

Example:

```
clear l2x statistics l2x-test1/0
```

# 4. Layer 3 Services

The Layer 3 Services section of the RtBrick FullStack (RBFS) User Guides details the configuration and management of network-layer virtualization, specifically focusing on the implementation of L3VPN services.

The following guide is available under the Layer 3 Services section:

- L3VPN

## 4.1. L3VPN

### 4.1.1. L3VPN Overview

L3VPN, or Layer 3 Virtual Private Network, is a type of MPLS VPN that operates on top of an IP/MPLS transport infrastructure. It allows customers to create secure and private layer 3(IP) connections over a shared infrastructure and provides seamless IP connectivity to the customer's multiple sites.

The key aspects of L3VPN are:

- **IP-Based Routing**: All the control plane communication in the L3VPN network is based on IP-based routing protocols, so there are dynamic updates, multihoming, fault detection, and optimal data routing.
- **Provider Edge (PE)**: The Layer 3 device at the edge of the service provider network is called Provider Edge.
- **Customer Edge (CE)**: The Layer 3 device at the edge of the customer network with its interface connected to the service provider is called the Customer Edge. Both PE and CE have IP connectivity between them, either through Routing Protocols or Static Routing. The PE device has a separate instance configured for each customer. An Instance is a virtual routing and forwarding table (VRF) along with customer-private routing protocols. The CE device is responsible for transferring all the routing information in the customer sites to the PE.
- **Encapsulation and Tunneling**: L3VPN uses the MPLS data plane encapsulation to securely transfer data between two PE routers in the service provider network.
- **VPN Instances**: Each customer has a separate instance (VPN routing and forwarding (VRF) ) in PE. This ensures that each customer's data remains separate and maintains data privacy.
- **Security and Privacy**: L3VPN, using the above techniques, such as Encapsulation, Tunneling, and VPN instances, provides isolation for each customer in the shared network.
- **Scalability and Flexibility**: L3VPN supports different network deployments depending on customer requirements. Instances are deployed only on the Edges, so they can be easily scaled. Filtering based on Extended Communities Route targets provides flexibility at the policy level, so it is possible to create various VPN topologies, such as hub-and-spoke VPNs, etc.

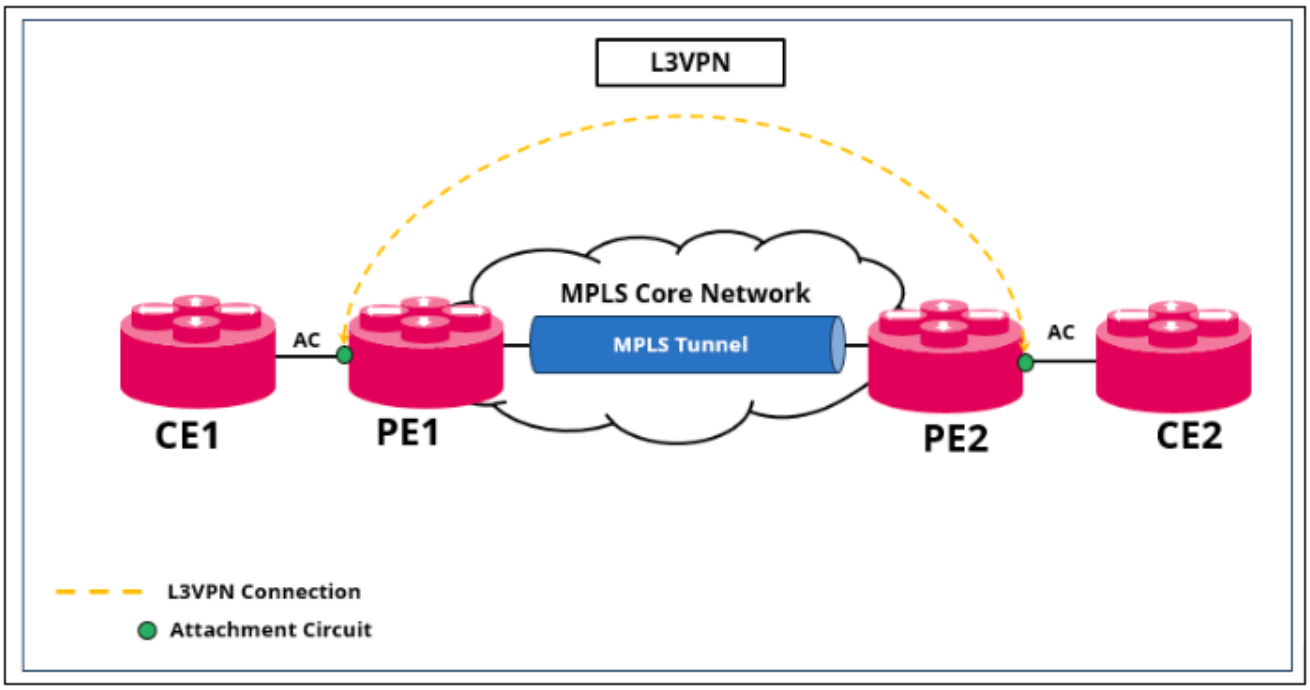



*Figure 54. L3VPN Overview*

PE1 and PE2 are each configured with a BGP L3VPN instance connected to the customer edge devices (CEs). Both PE and CE have IP connectivity between them, either through Routing Protocols or Static Routing. Once PEs receive customer routes, they use Multi-Protocol BGP (MP-BGP) to exchange the routes between the PEs.

For example, when CE1 needs to transmit packets to CE2, then CE1 initiates a route lookup process to determine the nexthop for getting to CE2. This destination route is exchanged in advance between PEs and CEs through the configured routing protocol. Upon obtaining the routing destination via PEs, the CEs proceed to transmit the packet to PEs. After the PEs receive packets from the CEs, they add the VPN labels which are advertised by the neighboring PEs for the destination route, and send them to the peer PEs using the MPLS label-switched path transport (LSP). After the other PEs receive the packet via the MPLS LSP, they pop the VPN labels of the packets and perform an IP route lookup on the L3VPN instances, and forward the packets to the destination CE2.

### Interprovider Layer 3 VPN options

Sometimes, a customer needs an L3 VPN between two locations where a given Service Provider (SP) does not have coverage. To interconnect these customers, Layer 3 VPNs can be deployed in three different ways: Option A, Option B, and Option C.

- **Option A**: When dealing with another ISP, use eBGP, but place the BGP session within the customer's VRF. Essentially, treat the other service provider as if they were just another customer CE. This approach requires each customer's individual BGP and VRF configurations. However, it is ideal when you have no control over the other ISP's network.
- **Option B**: When setting up eBGP with the other ISP, make sure to use the global/default routing table. Each ASBR should advertise a unique label for every destination FEC in every VRF to the other ASBR. This process is similar to creating a label-switched path between the two ISPs.
- **Option C**: Configure eBGP between the ASBRs again, but this time using two planes: 1) The VPN

unicast plane for exchanging VPN routes and plane 2 uses BGP Labeled Unicast, which enables PEs in one ISP to establish a label-switched path to the loopbacks of the PEs in the other ISP.

#### Supported L3VPN Standards

| RFC Number | Description |
|---|---|
| RFC 4364 | BGP/MPLS IP Virtual Private Networks (VPNs) |

RFC and draft compliance are partial except as specified.

#### Supported Platforms

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

### 4.1.2. L3VPN Configuration

#### Configuration Hierarchy

The diagram below shows the L3VPN configuration hierarchy.

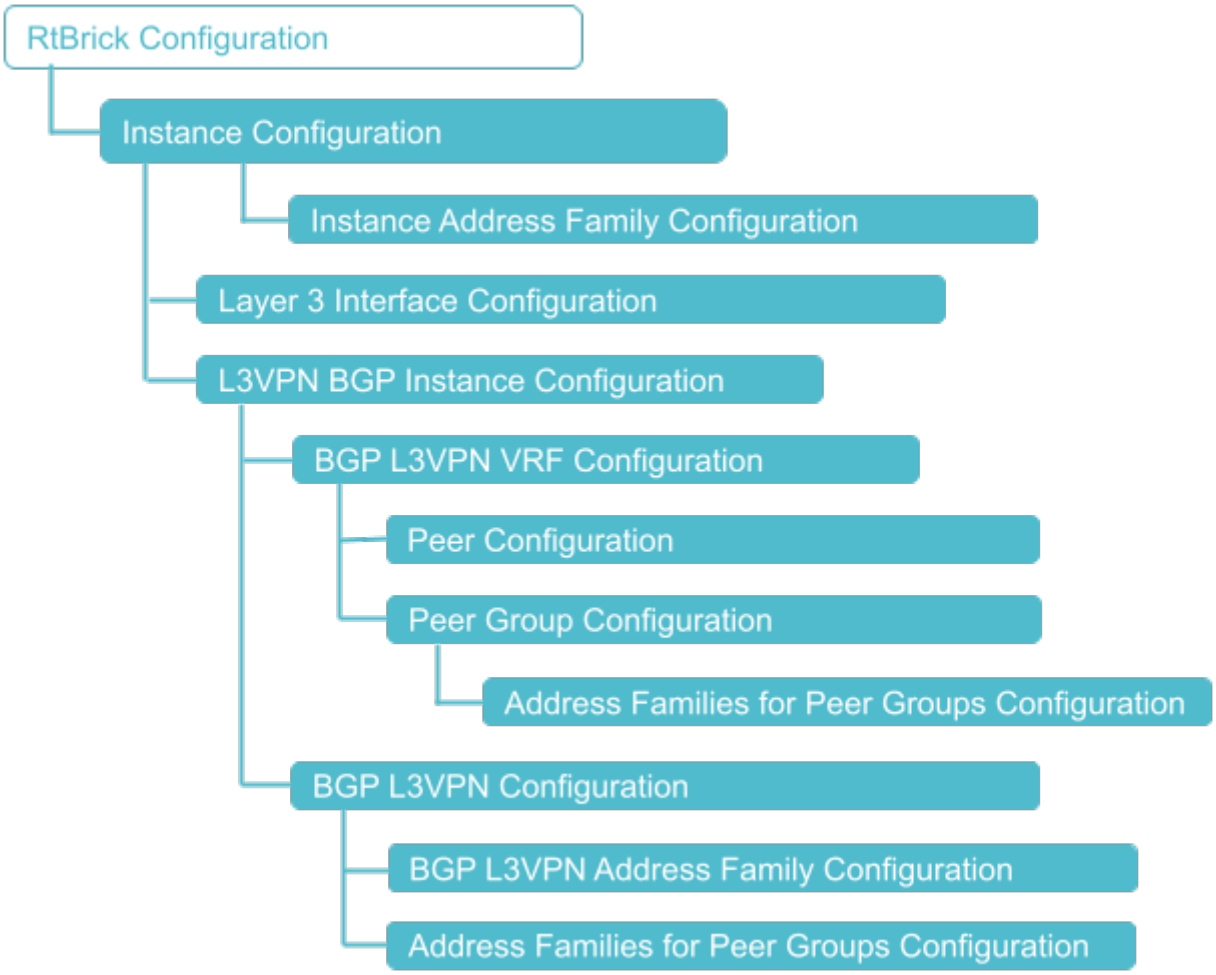



*Figure 55. L3VPN Configuration Hierarchy*

### Configuration Syntax and Commands

The following sections describe the BGP configuration syntax and commands.

#### Instance Configuration

Each VPN is associated with one or more VPN routing and forwarding (VRF) instances. A VRF defines the VPN membership of a customer site attached to a PE router. A VRF consists of the following components:

- An IP version 4 (IPv4) 6 (IPv6) unicast routing table
- A private FIB table
- A set of interfaces that use the forwarding table
- A set of rules and routing protocol parameters that control the information that is included in the routing table

These components are collectively called a VRF instance.

The instance configuration hierarchy includes parameters that are required for or used by BGP, but that are not part of the BGP protocol configuration hierarchy itself. Route distinguishers and router IDs are configured directly at the instance hierarchy.

**Syntax:**

**set instance** <instance-name> <attribute> <value>

| Attribute | Description |
|---|---|
| `<instance-name>` | Name of the routing instance |
| `route-distinguisher <as-number\|ipv4-address:id>` | The route distinguisher (RD) uniquely defines routes within an IPv4 network. PE routers use route distinguishers to identify which VPN a packet belongs to. Supported formats are <as-number:id> or <ipv4-address:id>.<br>If you want to use the format <as-number:id> with a 4-byte ASN, specify it with an "L". For example, set instance services route-distinguisher 4200000000L:101 |
| `ipv4-router-id <ipv4-address>` | The router ID of the routing instance. |

The following example configures tge instance identifier.

```
set instance l3vpn-ce1
set instance l3vpn-ce1 ipv4-router-id 172.16.3.10
set instance l3vpn-ce1 route-distinguisher 172.16.3.10:65001
commit
```

Example: Instance Identifier Configuration

```
supervisor@PE1: cfg> show config instance l3vpn-ce1
{
  "rtbrick-config:instance": [
    {
      "name": "l3vpn-ce1",
      "ipv4-router-id": "172.16.3.10",
      "route-distinguisher": "172.16.3.10:65001",
  <...>
```

#### Address Families

At the instance address family hierarchy, you can enable or disable address families for the instance, and configure parameters like route targets.

Please note default settings depend on the instance. For the 'default' instance, the IPv4 and IPv6 unicast, multicast, and labeled unicast, as well as the MPLS unicast address families are enabled by default. For any non-default instance, no address family is enabled by default and needs to be enabled by configuration.

**Syntax:**

**set instance** <instance-name> **address-family** <afi> <safi> <attribute> <value>

| Attribute | Description |
|---|---|
| `<afi>` | Address family identifier (AFI). Supported values: ipv4, ipv6 |
| `<safi>` | Subsequent address family identifier (SAFI). Supported values: unicast |
| `route-target ( import \| export ) <rt-value>` | Route targets (RT) are used to transfer routes between VPN instances. The RT identifies a subset of routes that should be imported to or exported from a particular VPN instance. You can configure an RT for importing or exporting routes or both.<br>If you want to use the format <as-number:id> with a 4-byte ASN, specify it with an "L". For example, set instance services address-family ipv4 unicast route-target export target:4200000000L:14 |
| `policy ( import \| export ) <policy-name>` | There are two attachment points for BGP policies. At this configuration hierarchy, you can attach import or export policies to the instance. These policies apply when routes are imported from the BGP protocol into the instance, or exported from the instance to the BGP protocol. |

The following example configures the instance address families.

```
set instance l3vpn-ce1 address-family ipv4 unicast
set instance l3vpn-ce1 address-family ipv4 unicast route-target import
target:172.16.3.10:65001
set instance l3vpn-ce1 address-family ipv4 unicast route-target export
target:172.16.3.10:65001
set instance l3vpn-ce1 address-family ipv6 unicast
set instance l3vpn-ce1 address-family ipv6 unicast route-target import
target:172.16.3.11:65001
set instance l3vpn-ce1 address-family ipv6 unicast route-target export
target:172.16.3.11:65001
commit
```

Example: Instance Address Family Configuration

```
supervisor@PE1: cfg> show config instance l3vpn-ce1 address-family
{
  "rtbrick-config:address-family": [
    {
      "afi": "ipv4",
      "safi": "unicast",
      "route-target": {
        "import": [
          "target:172.16.3.10:65001"
          ],
        "export": [
          "target:172.16.3.10:65001"
          ]
      }
    },
    {
      "afi": "ipv6",
      "safi": "unicast",
      "route-target": {
        "import": [
          "target:172.16.3.11:65001"
          ],
        "export": [
          "target:172.16.3.11:65001"
          ]
      }
    }
  ]
}
```

#### Layer 3 Interface Configuration

L3VPN supports the configuration of Layer 3 logical interfaces.

**Syntax:**

**set interface** <name> <attribute> <value>

| Attribute | Description |
|---|---|
| `unit <unit-id>` | Create a logical interface (also referred to as a sub-interface) under the physical interface. |
| `inner-vlan <inner-vlan-id>` | Inner VLAN ID. |
| `instance <instance>` | Assign the logical interface to an instance. |
| `vlan <outer-vlan-id>` | Outer VLAN ID. |

The following example shows the interface configuration.

```
set interface ifp-0/0/2 unit 0
set interface ifp-0/0/2 unit 0 instance l3vpn-ce1
set interface ifp-0/0/2 unit 0 address ipv4 10.0.0.1/24
commit
```

Example: Interface Configuration

```
supervisor@PE1: cfg> show config interface ifp-0/0/2
{
  "rtbrick-config:interface": [
    {
      "name": "ifp-0/0/2",
      "unit": [
        {
          "unit-id": 0,
          "instance": "l3vpn-ce1",
          "address": {
            "ipv4": [
              {
                "prefix4": "10.0.0.1/24"
              }
            ]
          }
        }
      ]
    }
  ]
}
```

#### Interface Address Configuration

This section describes how to configure interface IP addresses.

**Syntax:**

**set interface** <interface-name> **unit** <unit-id> **address** <afi> <attribute> <value>

| Attribute | Description |
|---|---|
| `<afi>` | Address family identifier (AFI). Supported values: ipv4 and ipv6 |
| `<prefix4\|prefix6>` | Assign IPv4 or IPv6 address to the interface unit. |

The following show the interface address configuration.

```
set interface ifp-0/0/2 unit 0
set interface ifp-0/0/2 unit 0 instance l3vpn-ce1
set interface ifp-0/0/2 unit 0 address ipv4 10.0.0.1/24
commit
```

Example: Interface Address Configuration

```
supervisor@PE1: cfg> show config interface ifp-0/0/2
{
  "rtbrick-config:interface": [
    {
      "name": "ifp-0/0/2",
      "unit": [
        {
          "unit-id": 0,
          "instance": "l3vpn-ce1",
          "address": {
            "ipv4": [
              {
                "prefix4": "10.0.0.1/24"
              }
            ]
          }
        }
      ]
    }
  ]
}
```

#### L3VPN BGP Instance Configuration

#### BGP L3VPN VFT (Virtual Forwarding Table) Configuration

**Syntax:**

**set instance** <name> **protocol bgp** <attribute> <value>

| Attribute | Description |
|---|---|
| `<name>` | Name of the routing instance |
| `host-name <host-name>` | The name of the BGP host, to a maximum of 64 characters |
| `domain-name <domain-name>` | The name of the BGP routing domain, to a maximum of 64 characters |
| `address-family <afi>` | Address family identifier (AFI). Supported value: ipv4, ipv6 |
| `address-family <afi> <safi>` | Subsequent address family identifier (SAFI). Supported value: unicast |
| `local-as <as-number>` | The AS number in four-byte format. The numbers allowed are from 1 to 4294967295. |
| `med <med-value>` | The BGP Multi-Exit Discriminator (MED) value. The numbers allowed are from 0 to 4294967295. When an AS has multiple links to another AS, the MED value is used to determine the exit to use to reach the other AS. |
| `router-id <router-id>` | Router identifier in IPv4 format |
| `timer hold-time <seconds>` | Hold timer in seconds. The valid range is 5 to 65535. |
| `timer keepalive <seconds>` | Keep a live timer in seconds. The valid range is 5 to 65535. |
| `redistribute <source>` | Enable the redistribution feature to dynamically inject specific types of routes into the BGP protocol. Supported route sources are `direct`, `igmp`, `ipoe`, `isis`, `ospf`, `pim`, `ppp`, `static`, and `arp-nd`. |
| `redistribute <source> policy <policy>` | Attach a policy to the redistribution process |

The following example configures BGP L3VPN VFT (Virtual Forwarding Table).

```
set instance l3vpn-ce1 protocol bgp domain-name PE1-AS1
set instance l3vpn-ce1 protocol bgp hostname PE1-AS1
set instance l3vpn-ce1 protocol bgp local-as 65001
set instance l3vpn-ce1 protocol bgp router-id 172.16.3.10
set instance l3vpn-ce1 protocol bgp address-family ipv4 unicast
set instance l3vpn-ce1 protocol bgp address-family ipv6 unicast
commit
```

Example: BGP L3VPN VFT (Virtual Forwarding Table)

```
supervisor@PE1: cfg> show config instance l3vpn-ce1 protocol bgp
```

```
{
  "rtbrick-config:bgp": {
    "domain-name": "PE1-AS1",
    "hostname": "PE1-AS1",
    "local-as": 65001,
    "router-id": "172.16.3.10",
    "address-family": [
      {
        "afi": "ipv4",
        "safi": "unicast"
      },
      {
        "afi": "ipv6",
        "safi": "unicast"
      }
    ],
 <...>
```

#### Peer Configuration

Once peer groups have been defined, BGP peers can be configured at the peer configuration hierarchy. A peer can be specified by address, or by interface when using IPv6 auto-discovered neighbors and link-local addresses. Furthermore, it is possible to configure TCP authentication and bind it to a peer.

Syntax to configure a BGP peer by address:

**set instance** <instance-name> **protocol bgp peer** ( ipv4 | ipv6) <peer-address> <update-source> **peer-group** <peer-group>

Syntax to configure a BGP peer using IPv6 link-local addresses:

**set instance** <instance-name> **protocol bgp peer interface** <name> **peer-group** <peer-group>

Syntax to configure TCP Authentication for BGP peers:

**set instance** <instance-name> **protocol bgp peer** (ipv4 | ipv6) <peer-address> <update-source> **authentication-id** <authentication-id>

| Attribute | Description |
|---|---|
| `interface <name>` | Enable BGP peer using IPv6 link-local addresses |
| `ipv4 <peer-address>` | IPv4 address of a BGP peer |
| `ipv6 <peer-address>` | IPv6 address of a BGP peer |

| Attribute | Description |
|---|---|
| `allow-as-in <value>` | Specify the value for allow-as-in. Allowed range of value 1 - 10. |
| `<update-source>` | Local IP address to be used for the peering |
| `peer-group <peer-group>` | Assign the peer to a peer group |
| `authentication-id <authentication-id>` | Authentication identifier |

The following example configures a peer.

```
set instance l3vpn-ce1 protocol bgp peer ipv4 10.0.0.9 10.0.0.1
set instance l3vpn-ce1 protocol bgp peer ipv4 10.0.0.9 10.0.0.1 peer-group ce1
commit
```

Example: Peer Configuration

```
supervisor@PE1: cfg> show config instance l3vpn-ce1 protocol bgp peer
{
  "rtbrick-config:peer": {
    "ipv4": [
      {
        "peer-address": "10.0.0.9",
        "update-source": "10.0.0.1",
        "peer-group": "ce1"
      }
    ]
  }
}
```

#### Peer Group Configuration

In BGP, neighbor peers with the same update policies can be grouped to simplify the initial configuration and updates. Peers share the same policies such as route maps, distribution lists, filter lists, update sources, and so on, so peer groups only need one configuration statement for these values.

**Syntax:**

**set instance** <instance-name> **protocol bgp peer-group** <peer-group-name> <attribute> <value>

| Attribute | Description |
|---|---|
| `local-as <as-number>` | Local AS number for the peer group |
| `remote-as <as-number>` | Remote AS number for the peer group |

| Attribute | Description |
|---|---|
| `any-as <true\|false>` | Enable dynamic AS negotiation for this peer group |
| `ebgp-multihop <hop-count>` | By default, the maximum number of hops between eBGP peers is 1 (direct connection). This hop count overrides the default behavior allowing connectivity between eBGP peers not directly connected. |

The following example configures a peer group.

```
set instance l3vpn-ce1 protocol bgp peer-group ce1
set instance l3vpn-ce1 protocol bgp peer-group ce1 remote-as 65009
commit
```

Example: Peer Group Configuration

```
supervisor@PE1: cfg> show config instance l3vpn-ce1 protocol bgp peer-group
{
  "rtbrick-config:peer-group": [
    {
      "pg-name": "ce1",
      "remote-as": 65009,
<...>
```

#### Configuring Address Families for Peer Groups

At this configuration hierarchy, you can enable the address families that shall be supported for the group peers, and enable features specific to the address family.

**Syntax:**

**set instance** <instance-name> **protocol bgp peer-group** <peer-group-name> **address-family** <afi> <safi> <attribute> <value>

| Attribute | Description |
|---|---|
| `<afi>` | Address family identifier (AFI). Supported values: ipv4, or ipv6 |
| `<safi>` | Subsequent address family identifier (SAFI). Supported values: unicast<br><br>` |
| `update-nexthop ( ipv4-address \| ipv6-address ) <address>` | BGP nexthop address for routes advertised to this peer group |
| `policy (import \| export ) <policy-name>` | Apply a routing policy to the peer group |

The following example configures the address families that shall be supported for the group peers

```
set instance l3vpn-ce1 protocol bgp peer-group ce1 address-family ipv4 unicast
set instance l3vpn-ce1 protocol bgp peer-group ce1 address-family ipv6 unicast
commit
```

Example: Peer Groups Address Family Configuration

```
supervisor@PE1: cfg> show config instance l3vpn-ce1 protocol bgp peer-group ce1
address-family
{
  "rtbrick-config:address-family": [
    {
      "afi": "ipv4",
      "safi": "unicast"
    },
    {
      "afi": "ipv6",
      "safi": "unicast"
    }
  ]
}
```

#### BGP L3VPN Configuration

##### Configuring the BGP L3VPN Address Families

**Syntax:**

**set instance** <name> **protocol bgp address-family** <afi> <safi>

| Attribute | Description |
|---|---|
| `<name>` | Name of the routing instance |
| `<afi>` | Address family identifier (AFI). Supported values: ipv4, ipv6 |
| `<safi>` | Subsequent address family identifier (SAFI). Supported values: vpn-unicast |

To configure BGP L3VPN on the default instance, enter the following command:

```
set instance default protocol bgp address-family ipv4 vpn-unicast
set instance default protocol bgp address-family ipv6 vpn-unicast
commit
```

Example: BGP L3VPN Address Family Configuration

```
supervisor@PE1: cfg> show config instance default protocol bgp address-family
{
  "rtbrick-config:address-family": [
    {
      "afi": "ipv4",
      "safi": "vpn-unicast"
    },
    {
      "afi": "ipv6",
      "safi": "vpn-unicast"
    }
  ]
}
```

#### Configuring Address Families for Peer Groups

**Syntax:**

**set instance** <instance-name> **protocol bgp peer-group** <pg-name> **address-family** <afi> <safi> <attribute> <value>

| Attribute | Description |
|---|---|
| `<afi>` | Address family identifier (AFI). Supported value: ipv4, ipv6. |
| `<safi>` | Subsequent address family identifier (SAFI). Supported value: vpn-unicast |
| `add-path` | Negotiate additional path capabilities with these peers, so that more than one path can be active to the peers in the group |
| `default-information originate <true\|false>` | Generate and advertise a default route to peers in the group |
| `extended-nexthop` | Enable extended-next-hop encoding for BGP peer groups to allow the transfer of IPv4 prefixes over an IPv6 connection |
| `nexthop-self <true\|false>` | Set the advertised BGP nexthop to yourself, this is the default for eBGP |
| `nexthop-unchanged <true\|false>` | Do not modify the advertised BGP nexthop, this is the default for iBGP |
| `update-nexthop ( ipv4-address \| ipv6-address ) <address>` | BGP nexthop address for routes advertised to this peer group |
| `remove-private-as <true\|false>` | Remove private AS numbers from routes advertised to group peers |
| `route-reflect-client <true\|false>` | Configure this peer as a route reflector client |
| `policy ( import \| export ) <policy-name>` | Apply a routing policy to the peer group |

To configure BGP L3VPN Peer Group on the `default` instance, enter the following command:

```
set instance default protocol bgp peer-group PE2-AS1 address-family ipv4 vpn-unicast
set instance default protocol bgp peer-group PE2-AS1 address-family ipv6 vpn-unicast
commit
```

Example: Configuring Address Families for Peer Groups

```
supervisor@PE1: cfg> show config instance default protocol bgp peer-group PE2-AS1
{
  "rtbrick-config:peer-group": [
    {
      "pg-name": "PE2-AS1",
      "remote-as": 4200000001,
      "address-family": [
        {
          "afi": "ipv4",
          "safi": "vpn-unicast"
        },
        {
          "afi": "ipv6",
          "safi": "vpn-unicast"
        }
      ]
    }
  ]
}
```

### 4.1.3. L3VPN Operational Commands

#### L3VPN Show Commands

##### BGP Summary

This command displays BGP protocol parameters like attributes or timers that are generic to the BGP instance.

**Syntax:**

**show bgp summary** <option>

| Attribute | Description |
|---|---|
| Option | Description |
| - | Without any option, the command displays the information for all instances. |

| Attribute | Description |
| --- | --- |
| `instance <instance-name>` | BGP summary information for the given instance. |

Example: BGP summary for L3VPN VRF instance

```
supervisor@rtbrick: op> show bgp summary instance ce1-vrf
Instance: ce1-vrf
  General information
    Hostname: ce1-vrf instance, Domain name: ce1-vrf instance
    Local AS: 64501, Version: 4
    Local preference: 100, eBGP Protocol preference: 20, iBGP Protocol preference: 200
    Router ID: 198.51.100.201, Cluster ID: 198.51.100.201
    MED: 0, Enforce first AS: True
    Address families configured:
      ipv4: unicast
      ipv6: unicast
  Capabilities
    Route refresh: True, AS4: True, Graceful restart: False,
  Best route selection
    Always compare MED: True, Ignore as path: False
    Ignore local preference: False, Ignore origin: False
    Ignore MED: False, Ignore route source: False
    Ignore router ID: False, Ignore uptime: True
    Ignore cluster length: False, Ignore peer IP: False
    Route select parameter: 0
  Timers
    Connect retry: 30s, Keepalive: 30s, Holdtime: 90s
  Statistics
    Peers configured: 0, Peers auto discovery: 0
      Peers in idle          : 0
      Peers in connect       : 0
      Peers in active        : 0
      Peers in opensent      : 0
      Peers in openconfirm   : 0
      Peers in established   : 0
```

Example: BGP summary for L3VPN default instance

```
supervisor@rtbrick: op> show bgp summary instance default
Instance: default
  General information
    Hostname: PE1, Domain name: Rtbrick
    Local AS: 65541, Version: 4
    Local preference: 100, eBGP Protocol preference: 20, iBGP Protocol preference: 200
    Router ID: 198.51.100.11, Cluster ID: 198.51.100.11
    MED: 0, Enforce first AS: True
    Address families configured:
      ipv4: labeled-unicast, vpn-unicast
```

```
      ipv6: labeled-unicast, vpn-unicast
  Capabilities
    Route refresh: True, AS4: True, Graceful restart: False,
  Best route selection
    Always compare MED: True, Ignore as path: False
    Ignore local preference: False, Ignore origin: False
    Ignore MED: False, Ignore route source: False
    Ignore router ID: False, Ignore uptime: True
    Ignore cluster length: False, Ignore peer IP: False
    Route select parameter: 0
  Timers
    Connect retry: 30s, Keepalive: 30s, Holdtime: 90s
  Statistics
    Peers configured: 1, Peers auto discovery: 0
      Peers in idle          : 0
      Peers in connect       : 0
      Peers in active        : 0
      Peers in opensent      : 0
      Peers in openconfirm   : 0
      Peers in established   : 1
```

#### BGP Peer

The 'show bgp peer' commands display information on BGP peers.

**Syntax:**

**show bgp peer** <option>

| Option | Description |
|---|---|
| - | Without any option, the commands display all BGP peers in all instances in a summary table format. |
| detail | Detailed information on all BGP peers in all instances in a list view. |
| <peer-name> | Detailed information on the peer with the given name. |
| address <peer-address> | Detailed information on the peer with the given IP address. |
| history | Displays BGP peer history information such as the peer state down reasons. |
| history <peer-address> | Displays BGP peer history information such as the peer state down reasons for a specified peer. |
| instance <instance-name> | Summary of all BGP peers in the given instance. |
| instance <instance-name> detail | Detailed information on all BGP peers in the given instance. |

| Option | Description |
|---|---|
| `instance <instance-name> detail <peer-name>` | Detailed information on the peer with the given name in the given instance. |
| `instance <instance-name> detail address <peer-address>` | Detailed information on the peer with the given IP address in the given instance. |
| `statistics` | Received and sent BGP prefixes per AFI/SAFI for all peers in all instances. |
| `statistics peer <peer-name>` | Received and sent BGP prefixes per AFI/SAFI for the peer with the given name. |
| `statistics peer address <peer-address>` | Received and sent BGP prefixes per AFI/SAFI for the peer with the given IP address. |
| `statistics instance <instance-name> peer <peer-name>` | Received and sent BGP prefixes per AFI/SAFI for the peer with the given name in the given instance. |
| `statistics instance <instance-name> peer address <peer-address>` | Received and sent BGP prefixes per AFI/SAFI for the peer with the given IP address in the given instance. |

Example: BGP Peer Information for L3VPN VRF instance

```
supervisor@rtbrick: op> show bgp peer instance l3vpn-ce1
Instance: l3vpn-ce1
  Peer                                    Remote AS    State         Up/Down Time              PfxRcvd
PfxSent
  CE1                                     65009        Established   0d:07h:04m:09s            7                  14
```

Example: BGP Peer Information for L3VPN default instance

```
supervisor@rtbrick: op> show bgp peer instance default
Instance: default
  Peer                                        Remote AS    State         Up/Down Time
PfxRcvd              PfxSent
  PE2                                         65542        Established   0d:00h:00m:13s
12                   20
```

Example: Detailed view of BGP Peer Information for L3VPN VRF instance

```
supervisor@rtbrick: op> show bgp peer instance l3vpn-ce1 detail
Peer: CE1, Peer IP: 10.0.0.9, Remote AS: 65009, Local: 10.0.0.1, Local AS: 65001, Any AS: False
  Type: ebgp, State: Established, Up/Down Time: Thu Dec 05 05:22:33 GMT +0000 2024, Reason: Cease, Sub-Code: Admin reset
  Discovered on interface: -
  Last transition: Thu Dec 05 05:22:33 GMT +0000 2024, Flap count: 1
  Peer ID        : 172.16.2.10, Local ID  : 172.16.3.10
  Instance       : l3vpn-ce1, Peer group: ce1
  6PE enabled    : False,
  TTL Security   : False, TTL Limit : -
  Timer values:
    Peer keepalive : 30s, Local keepalive: 30s
    Peer holddown  : 90s, Local holddown : 90s
```

```
    Connect retry  : 30s
  Timers:
    Connect retry timer : 0s
    Keepalive timer     : expires in 3s 266331us
    Holddown timer      : expires in 1m 26s 901069us
  NLRIs:
    Sent           : ['ipv4-unicast', 'ipv6-unicast']
    Received       : ['ipv4-unicast', 'ipv6-unicast']
    Negotiated     : ['ipv4-unicast', 'ipv6-unicast']
  Capabilities:
    Addpath sent                  : None
    Addpath received              : None
    Addpath negotiated            : None
    Extended nexthop sent         : None
    Extended nexthop received     : None
    Extended nexthop negotiated   : None
    Capabilities:
      Feature                      Sent            Received        Negotiated
      Route refresh                True            True            True
      4 byte AS                    True            True            True
      Graceful restart             False           False           False
      Link local only              False           False           False
  Prefix Limit:
  End of RIB:
    Address family                 Sent                                Received
    IPv4 unicast                   Thu Dec 05 05:22:38 GMT +0000 2024  Thu Dec 05 05:22:38 GMT +0000 2024
    IPv4 labeled-unicast           never                               never
    IPv6 unicast                   Thu Dec 05 05:22:38 GMT +0000 2024  Thu Dec 05 05:22:38 GMT +0000 2024
    IPv6 labeled-unicast           never                               never
    IPv4 VPN-unicast               never                               never
    IPv6 VPN-unicast               never                               never
    IPv4 flowspec                  never                               never
    IPv6 flowspec                  never                               never
    IPv4 VPN-multicast             never                               never
    L2VPN VPLS                     never                               never
    L2VPN EVPN                     never                               never
  RPKI Validation Stats:
    Address family                 Valid    Invalid    Unknown
    IPv4 unicast                       0          0          0
    IPv6 unicast                       0          0          0
  Message stats:
    Session stats:
      Direction              Open          Update       Keepalive          Notify   Route refresh
      Input                     1               4            1035               0               0
      Output                    1               9            1032               0               0
    Total stats:
      Input                     2               8            1101               0               0
      Output                    3              11            1097               1               0
    Route stats:
      Address family                       Received            Sent    Prefix limit    Idle timeout
      IPv4 unicast                                4               4               0               0
      IPv4 labeled-unicast                        0               0               0               0
      IPv6 unicast                                3               3               0               0
      IPv6 labeled-unicast                        0               0               0               0
      IPv4 VPN-unicast                            0               0               0               0
      IPv6 VPN-unicast                            0               0               0               0
      IPv4 VPN-multicast                          0               0               0               0
      L2VPN VPLS                                  0               0               0               0
      L2VPN EVPN                                  0               0               0               0
      IPv4 flowspec                               0               0               0               0
      IPv6 flowspec                               0               0               0               0
```

Example: Detailed view of BGP Peer Information for L3VPN default instance

```
supervisor@rtbrick: op> show bgp peer instance default detail
Peer: PE2, Peer IP: 192.0.2.2, Remote AS: 65542, Local IP: 192.0.2.1, Local AS: 65541,
Any AS: False
  Type: ebgp, State: Established, Up/Down Time: Wed Jul 15 19:01:01 GMT +0000 2026
  Discovered on interface: -
  Last transition: Wed Jul 15 19:01:01 GMT +0000 2026, Flap count: 0
  Peer ID        : 198.51.100.21, Local ID: 198.51.100.11
  Instance       : default, Peer group: PE2
  6PE enabled    : False
  TTL security   : False, TTL limit: -
  Authentication:
    ID: BGP_AUTH_HMAC-SHA-256-128, Algorithm: HMAC-SHA-256-128
  Timer values:
    Peer keepalive : 30s, Local keepalive: 30s
    Peer holddown  : 90s, Local holddown : 90s
    Connect retry  : 30s
  Timers:
    Connect retry timer : 0
    Keepalive timer     : expires in 8s 625937us
    Holddown timer      : expires in 1m 16s 126007us
  NLRIs:
    Sent           : ['ipv4-vpn-unicast', 'ipv6-vpn-unicast', 'ipv4-labeled-unicast',
'ipv6-labeled-unicast']
    Received       : ['ipv4-vpn-unicast', 'ipv6-vpn-unicast', 'ipv4-labeled-unicast',
'ipv6-labeled-unicast']
    Negotiated     : ['ipv4-vpn-unicast', 'ipv6-vpn-unicast', 'ipv4-labeled-unicast',
'ipv6-labeled-unicast']
  Capabilities:
    Addpath sent                  : None
    Addpath received              : None
    Addpath negotiated            : None
    Extended nexthop sent         : None
    Extended nexthop received     : None
    Extended nexthop negotiated   : None
    Multiple labels sent          : None
    Multiple labels received      : None
    Multiple labels negotiated    : None
    Capabilities:
      Feature                      Sent            Received        Negotiated
      Route refresh                True            True            True
      4 byte AS                    True            True            True
      Graceful restart             False           False           False
      Link local only              False           False           False
  Prefix Limit:
  End of RIB:
    Address family                 Sent                                Received
    IPv4 unicast                   never                               never
    IPv4 labeled-unicast           Wed Jul 15 19:01:07 GMT +0000 2026  Wed Jul 15
19:01:07 GMT +0000 2026
    IPv6 unicast                   never                               never
```

```
    IPv6 labeled-unicast           Wed Jul 15 19:01:07 GMT +0000 2026  Wed Jul 15
19:01:07 GMT +0000 2026
    IPv4 VPN-unicast               Wed Jul 15 19:01:07 GMT +0000 2026  Wed Jul 15
19:01:07 GMT +0000 2026
    IPv6 VPN-unicast               Wed Jul 15 19:01:07 GMT +0000 2026  Wed Jul 15
19:01:07 GMT +0000 2026
    IPv4 flowspec                  never                               never
    IPv6 flowspec                  never                               never
    IPv4 VPN-multicast             never                               never
    L2VPN VPLS                     never                               never
    L2VPN EVPN                     never                               never
  Message stats:
    Session stats:
      Direction              Open          Update       Keepalive          Notify
Route refresh
      Input                     1               8               1               0
0
      Output                    1               8               1               0
0
    Total stats:
      Input                     1               8               1               0
0
      Output                    1               8               1               0
0
    Route stats:
      Address family                       Received            Sent    Prefix limit
Idle timeout
      IPv4 unicast                                0               0               0
0
      IPv4 labeled-unicast                        4               6               0
0
      IPv6 unicast                                0               0               0
0
      IPv6 labeled-unicast                        4               6               0
0
      IPv4 VPN-unicast                            2               4               0
0
      IPv6 VPN-unicast                            2               4               0
0
      IPv4 multicast                              0               0               0
0
      IPv4 VPN-multicast                          0               0               0
0
      L2VPN VPLS                                  0               0               0
0
      L2VPN EVPN                                  0               0               0
0
      IPv4 Flowspec                               0               0               0
0
      IPv6 Flowspec                               0               0               0
```

```
0
```

#### BGP RIB-in

This command displays the received routes.

**Syntax:**

**show bgp rib-in** <option>

| Option | Description |
|---|---|
| - | Without any option, the command displays information on the received BGP routing table on all instances in a summary table format. |
| <afi> | BGP routing table summary for the given address family (AFI), all sub-address families and all instances. Supported AFI values are 'ipv4', 'ipv6' and 'l2vpn'. |
| <afi> <safi> | BGP routing table summary for the given address family (AFI) and sub-address family (SAFI), and all instances. Supported SAFI values are 'evpn', 'labeled-unicast', 'unicast', 'vpn-multicast', 'vpn-unicast', and 'vpls'. |
| <afi> <safi> detail | Detailed list view of the BGP routing table for the given address family (AFI) and sub-address family (SAFI), and all instances. |
| <afi> <safi> <prefix> | BGP routing table entry for the given prefix and all instances. |
| <afi> <safi> instance <instance-name> | BGP routing table summary for the given AFI, SAFI, and instance. |
| <afi> <safi> instance <instance-name> detail | Detailed list view of BGP routing table for the given AFI, SAFI, and instance. |
| <afi> <safi> instance <instance-name> <prefix> | BGP routing table entry for the given prefix and instance. |
| <afi> <safi> community <community-name> | BGP community details for the given AFI, SAFI, and instance. |
| <afi> <safi> error | BGP route with error status for the given AFI, SAFI, and instance. |
| <afi> <safi> peer <name> / peer address <ip> | Peer name or address |

Example: Summary view of the BGP rib-in where you can find received information for the specified L3VPN VRF instance.

```
supervisor@rtbrick: op> show bgp rib-in ipv4 unicast
Instance: ce1-vrf, AFI: ipv4, SAFI: unicast
  Hostname: Local, Peer IP: 0.0.0.0
  Source IP: 0.0.0.0, Total routes: 2
    Prefix                                        Next Hop
```

```
MED         Lpref       AS Path
    198.51.100.201/32                           -                                   0
-           -
    198.51.100.202/32                           -                                   0
-           -
```

Example: Summary view of the BGP rib-in where you can find received information for the specified L3VPN VRF instance of IPv6 prefixes.

```
supervisor@rtbrick: op> show bgp rib-in ipv6 unicast
Instance: ce1-vrf, AFI: ipv6, SAFI: unicast
  Hostname: Local, Peer IP: ::
  Source IP: ::, Total routes: 2
    Prefix                                      Next Hop
MED         Lpref       AS Path
    2001:db8:0:1::201/128                       -                                   0
-           -
    2001:db8:0:1::202/128                       -                                   0
-           -
```

Example: Summary view of the BGP rib-in where you can find received information of L3VPN default instance of address-family IPv4 vpn-unicast

```
supervisor@rtbrick: op> show bgp rib-in ipv4 vpn-unicast
Instance: default, AFI: ipv4, SAFI: vpn-unicast
  Hostname: PE2, Peer IP: 192.0.2.2
  Source IP: 192.0.2.1, Total routes: 2
    Prefix                                      Next Hop
MED         Lpref       AS Path
    198.51.100.221/32                           198.51.100.21                       0
-           65542
    198.51.100.222/32                           198.51.100.21                       0
-           65542
```

Example: Summary view of the BGP rib-in where you can find received information of L3VPN default instance of address-family IPv6 vpn-unicast.

```
supervisor@rtbrick: op> show bgp rib-in ipv6 vpn-unicast
Instance: default, AFI: ipv6, SAFI: vpn-unicast
  Hostname: PE2, Peer IP: 192.0.2.2
  Source IP: 192.0.2.1, Total routes: 2
    Prefix                                      Next Hop
MED         Lpref       AS Path
    2001:db8:0:1::221/128                       198.51.100.21                       0
-           65542
    2001:db8:0:1::222/128                       198.51.100.21                       0
```

```
-           65542
```

#### BGP FIB

The 'show bgp fib' commands display the BGP forwarding table.

**Syntax:**

**show bgp fib** <option>

| Option | Description |
|---|---|
| - | Without any option, the commands display the BGP forwarding table for all address families and all instances in a summary table format. |
| <afi> | BGP forwarding table summary for the given address family (AFI), all sub-address families and all instances. Supported AFI values are 'l2vpn', 'ipv4' and 'ipv6'. |
| <afi> <safi> | BGP forwarding table summary for the given address family (AFI) and sub-address family (SAFI), and all instances. Supported SAFI values are 'unicast', 'labeled-unicast', 'vpn-multicast', 'vpn-unicast', 'evpn-vpws', and 'vpls-vpws'. |
| <afi> <safi> detail | Detailed list view of the BGP forwarding table for the given address family (AFI) and sub-address family (SAFI), and all instances. |
| <afi> <safi> <prefix> | BGP forwarding table entry for the given prefix and all instances. |
| <afi> <safi> instance <instance-name> | BGP forwarding table summary for the given AFI, SAFI, and instance. |
| <afi> <safi> instance <instance-name> detail | Detailed list view of BGP forwarding table for the given AFI, SAFI, and instance. |
| <afi> <safi> instance <instance-name> <prefix> | BGP forwarding table entry for the given prefix and instance. |

Example: Summary view of the BGP FIB

```
supervisor@rtbrick: op> show bgp fib
Instance: ce1-vrf, AFI: ipv4, SAFI: unicast, Total routes: 2
  Prefix                                        Preference      Label
Next Hop
  198.51.100.221/32                             20              519938,bos:1
198.51.100.21
  198.51.100.222/32                             20              519938,bos:1
198.51.100.21
Instance: default, AFI: ipv4, SAFI: labeled-unicast, Total routes: 2
  Prefix                                        Preference      Label
```

```
Next Hop
  198.51.100.21/32                              20              -
192.0.2.2
  198.51.100.22/32                              20              -
192.0.2.2
Instance: default, AFI: ipv4, SAFI: vpn-unicast, Total routes: 4
  Prefix                                        Preference      Label
Next Hop
  198.51.100.201/32                             200             519938,bos:1         -
  198.51.100.202/32                             200             519938,bos:1         -
  198.51.100.221/32                             20              519938,bos:1
198.51.100.21
  198.51.100.222/32                             20              519938,bos:1
198.51.100.21
Instance: ce1-vrf, AFI: ipv6, SAFI: unicast, Total routes: 2
  Prefix                                        Preference      Label
Next Hop
  2001:db8:0:1::221/128                         20              519939,bos:1
198.51.100.21
  2001:db8:0:1::222/128                         20              519939,bos:1
198.51.100.21
Instance: default, AFI: ipv6, SAFI: labeled-unicast, Total routes: 2
  Prefix                                        Preference      Label
Next Hop
  2001:db8:0:1::21/128                          20              -
192.0.2.2
  2001:db8:0:1::22/128                          20              -
192.0.2.2
Instance: default, AFI: ipv6, SAFI: vpn-unicast, Total routes: 4
  Prefix                                        Preference      Label
Next Hop
  2001:db8:0:1::201/128                         200             519939,bos:1         -
  2001:db8:0:1::202/128                         200             519939,bos:1         -
  2001:db8:0:1::221/128                         20              519939,bos:1
198.51.100.21
  2001:db8:0:1::222/128                         20              519939,bos:1
198.51.100.21
```

#### BGP RIB-out

This command displays the routes that were advertised to peers.

**Syntax:**

**show bgp rib-out** <option>

| Option | Description |
| --- | --- |
| - | Without any option, the command displays advertised BGP routes for all instances. |
| <afi> | BGP routing table summary for the given address family (AFI), all sub-address families and all instances. Supported AFI values are 'l2vpn', 'ipv4' and 'ipv6'. |
| <afi> <safi> | BGP routing table summary for the given address family (AFI) and sub-address family (SAFI), and all instances. Supported SAFI values are 'evpn', 'unicast', 'labeled-unicast', 'multicast', 'vpn-unicast', and 'vpls'. |
| <afi> <safi> detail | Detailed list view of the BGP routing table for the given address family (AFI) and sub-address family (SAFI), and all instances. |
| <afi> <safi> <prefix> | BGP routing table entry for the given prefix and all instances. |
| <afi> <safi> instance <instance-name> | BGP routing table summary for the given AFI, SAFI, and instance. |
| <afi> <safi> instance <instance-name> detail | Detailed list view of BGP routing table for the given AFI, SAFI, and instance. |
| <afi> <safi> instance <instance-name> <prefix> | BGP routing table entry for the given prefix and instance. |
| <afi> <safi> peer <name> / peer address <ip> | Peer name or address |

Example: Summary view of the routes advertised to a L3VPN VRF instance peer of IPv4

```
supervisor@rtbrick: op> show bgp rib-out ipv4 vpn-unicast
Flags: N - RPKI Unknown, I - RPKI Invalid, V - RPKI Valid
Instance: default, AFI: ipv4, SAFI: vpn-unicast
  Peer-group: PE2, Sent routes: 4
    Flags  Prefix                                        MED     Lpref      Origin
Next Hop                                AS Path
           198.51.100.201/32                             0       -          Incomplete
198.51.100.11                           65541
           198.51.100.202/32                             0       -          Incomplete
198.51.100.11                           65541
           198.51.100.221/32                             0       -          Incomplete
198.51.100.11                           65541, 65542
           198.51.100.222/32                             0       -          Incomplete
198.51.100.11                           65541, 65542
```

Example: Summary view of the routes advertised to a L3VPN VRF instance peer of IPv6

```
supervisor@rtbrick: op> show bgp rib-out ipv6 unicast
Instance: l3vpn-ce1, AFI: ipv6, SAFI: unicast
  Peer-group: ce1, Sent routes: 6
    Prefix                                        MED     Lpref      Origin          Next Hop
```

```
      AS Path
    172:16:2::10/128                            0       -          Incomplete      -
      [65001, 65009]
    172:16:2::11/128                            0       -          Incomplete      -
      [65001, 65009]
    172:16:2::12/128                            0       -          Incomplete      -
      [65001, 65009]
    172:16:2::20/128                            0       -          Incomplete      -
      [65001, 4200000002, 65029]
    172:16:2::21/128                            0       -          Incomplete      -
      [65001, 4200000002, 65029]
    172:16:2::22/128                            0       -          Incomplete      -
      [65001, 4200000002, 65029]
```

Example: Summary view of the routes advertised to L3VPN default instance of address-family IPv4 vpn-unicast

```
supervisor@rtbrick: op> show bgp rib-out ipv4 vpn-unicast
Instance: default, AFI: ipv4, SAFI: vpn-unicast
  Peer-group: PE2, Sent routes: 8
    Prefix                                      MED     Lpref      Origin          Next Hop
      AS Path
    10.0.0.0/24                                 0       -          Incomplete      -
      [4200000001, 65009]
    10.0.1.0/24                                 0       -          Incomplete      -
      [4200000001, 4200000002, 65029]
    172.16.2.10/32                              0       -          Incomplete      -
      [4200000001, 65009]
    172.16.2.11/32                              0       -          Incomplete      -
      [4200000001, 65009]
    172.16.2.12/32                              0       -          Incomplete      -
      [4200000001, 65009]
    172.16.2.20/32                              0       -          Incomplete      -
      [4200000001, 4200000002, 65029]
    172.16.2.21/32                              0       -          Incomplete      -
      [4200000001, 4200000002, 65029]
    172.16.2.22/32                              0       -          Incomplete      -
      [4200000001, 4200000002, 65029]
```

Example: Summary view of the routes advertised to L3VPN default instance of address-family IPv6 vpn-unicast

```
supervisor@rtbrick: op> show bgp rib-out ipv6 vpn-unicast
Flags: N - RPKI Unknown, I - RPKI Invalid, V - RPKI Valid
Instance: default, AFI: ipv6, SAFI: vpn-unicast
  Peer-group: PE2, Sent routes: 4
    Flags  Prefix                                       MED     Lpref      Origin
Next Hop                                AS Path
           2001:db8:0:1::201/128                        0       -          Incomplete
::ffff:198.51.100.11                    65541
```

```
            2001:db8:0:1::202/128                         0       -          Incomplete
::ffff:198.51.100.11                     65541
            2001:db8:0:1::221/128                         0       -          Incomplete
::ffff:198.51.100.11                     65541, 65542
            2001:db8:0:1::222/128                         0       -          Incomplete
::ffff:198.51.100.11                     65541, 65542
```

# 5. Subscriber Management

Subscriber Management is purpose-built for access networks, delivering a scalable, distributed architecture that efficiently manages large volumes of subscriber sessions and AAA (authentication, authorization, and accounting) services.

It supports PPPoE, IPoE, and L2TP access types, seamlessly integrating with RADIUS-based AAA systems to enable flexible onboarding, streamlined service activation, dynamic policy enforcement, and comprehensive accounting.

Its modular architecture ensures effortless scaling of subscriber services, high availability, operational simplicity, and broad compatibility with industry-standard protocols.

Key Features:

- Centralized Authentication, Authorization and Accounting with RADIUS
- Dynamic subscriber provisioning and policy management
- RADIUS Accounting, Change of Authorization, and Disconnect support
- High scalability and resiliency with modular service components
- Support for 1:1 VLAN and N:1 shared VLAN models
- Flexible integration with carrier-grade broadband networks
- Local and remote subscriber management
- Carrier-Grade NAT (CGNAT)
- Redundancy

The following guides are available under the Subscriber Management section:

- Subscriber Management
- RBFS RADIUS Services
- RBFS CGNAT
- RBFS CGNAT Appliance
- RBFS HTTP Redirect Service
- RBFS Subscriber Filters
- Lawful Interception
- L2BSA
- IPoE Subscriber Redundancy
- PPPoE Subscriber Redundancy

## 5.1. Subscriber Management

### 5.1.1. Subscriber Management Overview

RtBrick's modular and scalable subscriber management, referred to as next-generation access infrastructure (ng-access), supports a range of access protocols including PPPoE, L2TPv2, IPoE (DHCPv4/v6), and RADIUS. This system is designed to deliver carrier-grade internet access services, with a focus on scalability, interoperability, and robustness. By providing support for multiple access protocols, RBFS ensures flexibility in deployment and compatibility with various network configurations, which is crucial for operators looking to provide consistent and high-quality services to their customers.

In this context, a subscriber refers to a customer connected through a Broadband Network Gateway (BNG). The BNG manages the connection by handling various protocols and sessions. Each subscriber is associated with a unique MAC address within a VLAN.

Providers that use separate VLANs for different services (e.g. internet and voice) may create multiple independent subscribers for the same customer or household. Although these subscribers might share certain QoS hierarchies, they are otherwise completely independent. Usually, VLANs used for services like IPTV or voice do not require subscriber management using static L3 interfaces with a traditional DHCP relay (non-subscriber).

A subscriber may have different forms of connectivity, depending on the access method used, such as PPPoE or IPoE. Thus, subscriber management involves overseeing customer connections, ensuring proper authentication, authorization, and accounting (AAA) of each subscriber, and maintaining the smooth operation of these sessions within a BNG. Effective subscriber management is key to maintaining high levels of customer satisfaction, as it directly impacts the reliability and performance of the internet services provided.

RBFS supports the following different protocol combinations to tailor an access network:

- **PPPoE Subscriber:** PPPoE/PPP Session and DHCPv6 (Prefix Delegation)
- **L2TP Subscriber:** PPPoE/PPP Session and L2TP Session
- **IPoE Subscriber:** DHCPv4 or DHCPv6

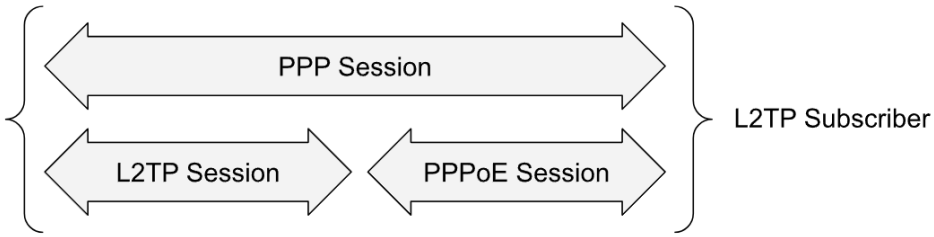



*Figure 56. Different Protocol Combinations*

RtBrick's implementation utilizes a distributed architecture to enhance both scalability and stability. The distributed nature of this architecture allows multiple processes to manage different aspects of subscriber connections concurrently, reducing bottlenecks and ensuring that the system can handle a large number of subscribers efficiently. Carrier networks often need to interoperate with a wide variety of client devices from different vendors, each requiring adherence to established industry standards and RFCs. This complexity necessitates a robust and flexible approach to managing different network interactions.

Furthermore, these networks must cope with the diverse behaviors of numerous access devices, some of which may have non-standard implementations but cannot be easily replaced due to logistical or cost-related constraints. This adds to the complexity of subscriber management, as non-standard devices may behave unpredictably, requiring additional handling to maintain stable connections. To address these challenges, RBFS provides a protocol stack specifically tailored to offer best-in-class interoperability, ensuring reliable connectivity across a diverse range of client devices. This approach not only ensures that the network can interoperate with devices from various vendors but also allows for customization and adaptation to unique network scenarios.

Overall, RtBrick's subscriber management solution is designed to meet the needs of modern carrier networks by providing a flexible, scalable, and reliable platform for managing customer connections. Its distributed architecture, combined with support for a wide array of protocols and strong adherence to industry standards, makes it well-suited to handle the challenges posed by diverse client environments and evolving network requirements. This ensures that network operators can provide a seamless and high-quality experience to their customers, even in the face of varying device capabilities and non-standard behaviors.

### Subscriber Management Daemons

There are four main daemons in the RtBrick distributed access architecture:

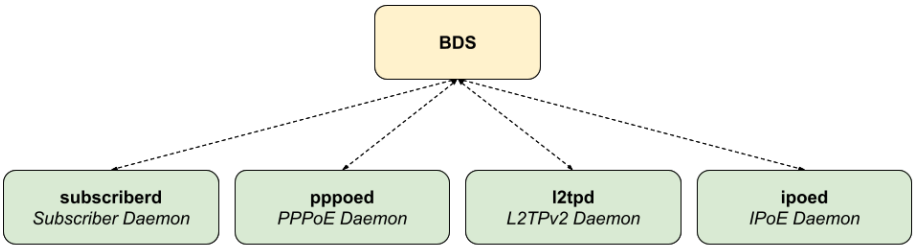



*Figure 57. RBFS Access Infrastructure*

The subscriber daemon (`subscriberd`) serves as the central component, managing the current subscriber state and handling Authentication, Authorization, and Accounting (AAA). The roles of each daemon are as follows:

- *subscriberd*: The subscriber daemon is responsible for subscriber management and AAA, which can be performed locally, via RADIUS, or through other methods. It also serves as the interface to the forwarding infrastructure, responsible to request per-subscriber QoS, filters, and more.
- *pppoed*: The PPPoE daemon handles PPPoE and PPP sessions for subscribers including subprotocols as ICMPv6 and DHCPv6 where the implemenation for PPPoE differs compared to IPoE.
- *l2tpd*: The L2TP daemon is responsible to manage L2TPv2 tunnels and sessions.
- *ipoed*: The IPoE (IP-over-Ethernet) daemon handles protocols related to IPoE subscribers like DHCPv4 and DHCPv6.

### Subscriber Identifier

Each subscriber in RBFS is uniquely identified by a 64-bit number called `subscriber-id`, which remains unique across all RBFS processes and even across chassis in the context of high availability.

This uniqueness is achieved through the following format:

- The first 4 bits identify the chassis, starting from 0.
- The next 4 bits identify the process:
  - 0x00 subscriberd (e.g. L2BSA, test, ...)
  - 0x01 pppoed (PPPoE sessions)
  - 0x02 l2tpd (reserved for LNS)
  - 0x03 ipoed (DHCPv4/v6)
- The following 8 bits are used to identify the application sharding instance. For example, instances such as `pppoed.1`, `pppoed.2`, and so on. While currently not in use, these bits provide the ability to partition load across multiple instances of the same process if needed to enhance scalability.
- The remaining 48 bits uniquely identify the subscriber, allowing for up to 281,474,976,710,656 unique subscribers per application instance before repetition, but even in the unlikely case of rollover, RBFS ensures uniqueness.

The subscriber portion always starts at '1', meaning that after a reboot, the subscriber identifiers will begin from the same value. These identifiers are consistent across multiple BNG, unless the `chassis-id` is explicitly set, which is required only for stateful high-availability configurations. Therefore, this number should not be used as a globally unique, non-repeating identifier in customer databases or management systems.

### VLAN Modes and Encapsulations

RBFS supports both 1:1 VLAN (VLAN-per-subscriber) and N:1 VLAN (shared VLAN) models for PPPoE and IPoE subscribers. It supports various VLAN encapsulations. Subscriber VLANs are dynamically created by specifying VLAN ranges with corresponding configurations. After successful authentication, RBFS generates the subscriber IFL, which includes the dynamic VLAN.

#### 1:1 VLAN Model (VLAN-per-subscriber)

In the 1:1 VLAN model, each subscriber is assigned a unique VLAN. This provides a direct one-to-one mapping between VLANs and subscribers. New services can be added easily without affecting others. The 1:1 model is recommended because it is simple and offers flexibility for features like backhauling, bitstream access, and QoS. As a result, more providers are migrating from the N:1 to the 1:1 VLAN model.

The following diagram illustrates a dedicated customer VLAN for each subscriber, showing the one VLAN per subscriber setup.

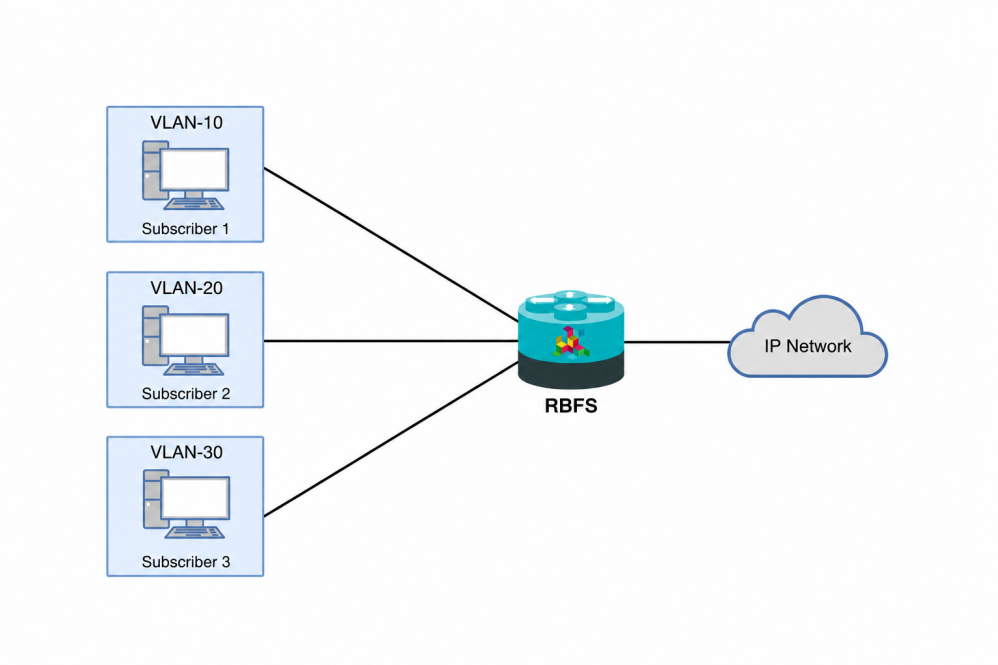



*Figure 58. Dedicated Customer VLAN for Each Subscriber*

This model is defined by the maximum number of subscribers per VLAN in the access-interface configuration, which has a default value of 1. As a result, the 1:1 VLAN model is also the default in RBFS.

#### N:1 VLAN Model (shared-VLAN)

The shared VLAN model offers many-to-one (N:1) subscriber-to-service connectivity, where multiple subscribers share a single VLAN. Unlike the 1:1 model, where each subscriber has a dedicated VLAN, the N:1 model provides a shared VLAN for many subscribers. A drawback of the shared VLAN model is the lack of logical isolation between subscribers at the VLAN level.

This results in certain limitations for IPoE subscribers, such as the lack of support for features like HTTP redirect, subscriber ACL, NAT, and IGMP for IPoE N:1 subscribers.

The following diagram shows a single VLAN that is connected to many subscribers.

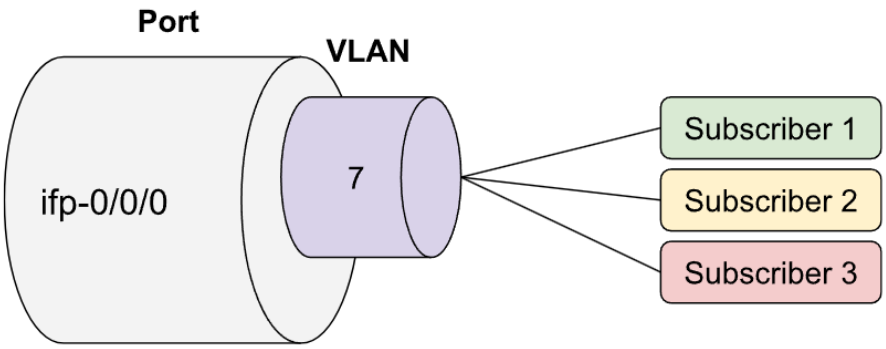



*Figure 59. Single VLAN that is Connected to Many Subscribers*

This model is defined by the maximum number of subscribers per VLAN in the access-interface configuration, which defaults to 1. Increasing this value beyond 1 enables the N:1 VLAN model for the corresponding access-interface.

### Auto-sensing VLAN TPID

In modern telecommunications networks, managing traffic for wholesale partners presents challenges, especially when the network operator lacks knowledge of the specific TPID (Tag Protocol Identifier) used in arriving traffic. This can lead to inefficiencies in network handling and potential service disruptions. To address this issue, RBFS supports Auto-sensing VLAN TPID which automatically detects and processes traffic tagged with different TPIDs. Specifically, this functionality will support both TPID 0x8100, commonly referred to as 802.1q, and TPID 0x88a8, referred to as 802.1ad.

The auto-sensing mechanism will utilize the initial subscriber-initiated packet, such as PPPoE Discovery packets (PADI, PADR), DHCP Discovery, or DHCPv6 Solicit packets, to determine the appropriate TPID. This ensures that the system can adapt to the tagging method used by the incoming traffic without requiring prior knowledge.

The default behavior is auto-sensing the TPID based on the first incoming packet, offering maximum flexibility. However, network operators will still have the option to explicitly define the `outer-vlan-encapsulation` in the access-interface configuration if required, overriding the auto-sensing functionality when needed. This approach enhances the scalability and efficiency of handling wholesale traffic across various network environments.

This applies to double-tagged profiles only! For single-tagged profiles, the default is 802.1q and cannot be changed, while for untagged profiles, no TPID check is performed as there is no VLAN present.

### Authentication, Authorization, and Accounting Using RADIUS

Remote Authentication Dial-In User Service (RADIUS) is a protocol that provides centralized Authentication, Authorization, and Accounting (AAA) for all types of subscribers. RADIUS servers can perform as authentication and accounting servers. Authentication servers maintain authentication records for subscribers.

The subscriber daemon optionally requests authentication using RADIUS access request messages before permitting subscribers access. Accounting servers handle accounting records for subscribers. The subscriber daemon transmits RADIUS accounting-on, start, interim, and stop messages to the servers. Accounting is the process of tracking subscriber activity and network resource usage in a subscriber session. This includes the session time called time accounting and the number of packets and bytes transmitted during the session called volume accounting. A RADIUS server can behave as a change of authorization (CoA) client, allowing dynamic changes for subscriber sessions. The subscriber daemon supports both RADIUS CoA messages and disconnects messages. CoA messages can modify the characteristics of existing subscriber sessions without loss of service, disconnect messages can terminate subscriber sessions. Each RADIUS request from the subscriber daemon includes the RADIUS accounting-session-id attribute (type 44) with a format that is configurable in the AAA configuration profile and includes at least the subscriber-id to identify the corresponding subscriber. The default format (<subscriber-id>.<timestamp>) includes also a Unix timestamp to ensure that the tuple of NAS-Identifier (e.g. hostname) and Accounting-Session-Id is global and unique to be usable as a key in RADIUS databases.

Additionally, to `subscriber-id` and `accounting-session-id` each subscriber consists also of a `subscriber-ifl` build based on physical port information and `subscriber-id` (ifp: ifp-0/0/1 and

subscriber-id: 72339069014638610 → subscriber-ifl: ppp 0/0/1/72339069014638610) which is required as a handle in the RBFS forwarding infrastructure.

```
    Code: Access-Request (1)
    Packet identifier: 0x22 (34)
    Length: 416
    Authenticator: e61a0dd74c74704f608688b08de1dfba
    [The response to this request is in frame 12]
  ▼ Attribute Value Pairs
    ▶ AVP: t=User-Name(1) l=19 val=user1@rtbrick.com
    ▶ AVP: t=CHAP-Challenge(60) l=18 val=2f696f4e920b47cab869021feb2bf632
    ▶ AVP: t=CHAP-Password(3) l=19 val=02f439040e9feb7bbc9e7622a364344913
    ▶ AVP: t=NAS-IP-Address(4) l=6 val=1.1.1.1
    ▶ AVP: t=NAS-Identifier(32) l=5 val=BNG
    ▶ AVP: t=NAS-Port-Id(87) l=59 val=BNG#hostif-0/0/4#10#7#0.0.0.0/0.0.0.0 eth 1#DEU.RTBRICK.1
    ▶ AVP: t=NAS-Port(5) l=6 val=67149831
    ▶ AVP: t=NAS-Port-Type(61) l=6 val=Ethernet(15)
    ▶ AVP: t=Service-Type(6) l=6 val=Framed(2)
    ▶ AVP: t=Framed-Protocol(7) l=6 val=PPP(1)
    ▶ AVP: t=Acct-Session-Id(44) l=30 val=72339069014638895:1589876315
    ▶ AVP: t=Vendor-Specific(26) l=13 vnd=RtBrick Inc.(50058)
    ▶ AVP: t=Vendor-Specific(26) l=20 vnd=RtBrick Inc.(50058)
    ▶ AVP: t=Vendor-Specific(26) l=16 vnd=RtBrick Inc.(50058)
    ▶ AVP: t=Vendor-Specific(26) l=25 vnd=RtBrick Inc.(50058)
    ▼ AVP: t=Vendor-Specific(26) l=16 vnd=RtBrick Inc.(50058)
        Type: 26
        Length: 16
        Vendor ID: RtBrick Inc. (50058)
      ▶ VSA: t=RtBrick-Subscriber-Id(25) l=10 val=010100000000012f
    ▼ AVP: t=Vendor-Specific(26) l=35 vnd=RtBrick Inc.(50058)
        Type: 26
        Length: 35
        Vendor ID: RtBrick Inc. (50058)
      ▶ VSA: t=RtBrick-Subscriber-Ifl(26) l=29 val=ppp-0/0/4/72339069014638895
    ▶ AVP: t=Vendor-Specific(26) l=29 vnd=The Broadband Forum(3561)
    ▶ AVP: t=Calling-Station-Id(31) l=23 val=0.0.0.0/0.0.0.0 eth 1
    ▶ AVP: t=Vendor-Specific(26) l=21 vnd=The Broadband Forum(3561)
    ▶ AVP: t=Vendor-Specific(26) l=18 vnd=The Broadband Forum(3561)
```

*Figure 60. RADIUS Access-Request*

> The subscriber-id is an unsigned 64-bit integer which is shown as a hex number in Wireshark.

Each subscriber is formed based on configuration profiles and individual settings retrieved via RADIUS. Conflicts between RADIUS-defined attributes and profile attributes are solved by prioritizing those received from RADIUS which is common best practice for broadband access concentrators. New subscribers are signalled via RADIUS access request and either accepted by RADIUS access-accept or rejected by RADIUS access-reject message from the RADIUS server. The RADIUS access-accept includes all attributes required to form the subscriber like IP addresses, DNS servers, and referenced configuration profiles. Some of those attributes can be changed by RADIUS dynamically using CoA requests without disconnecting the subscriber.

#### RADIUS Accounting

A RADIUS Acct-Status-Type attribute is used by the RADIUS client (subscriber daemon) to mark the start of accounting (for example, upon booting) by specifying Accounting-On and to mark the end of accounting (for example, just before a scheduled reboot) by specifying Accounting-Off. This message is often used by RADIUS servers to automatically close/terminate all open accounting records/sessions for the corresponding client, and therefore must not be sent to servers belonging to a profile that was already used/started for accounting.

Per default, the assumption is that all servers referenced by a RADIUS profile share the same states and therefore accounting-on must be only sent to one of those before the first accounting-start is

sent.

RADIUS `Accounting-On/Off` messages are optionally enabled in the RADIUS Profile Configuration using the `accounting-on-off` attribute. The additional attribute `accounting-on-wait` prevents any new session until accounting has started meaning that the `Accounting-On` response is received.

> Currently, `Accounting-Off` is not implemented.

RADIUS accounting requests are often used for billing and therefore should be able to store and retry over a more extended period (commonly up to 24 hours or more) which can be optionally enabled in the RADIUS profile configuration using the `accounting-backup` attribute. The maximum backup accounting hold time in seconds is defined in the attribute `accounting-backup-max`.

#### RADIUS Redundancy

RBFS allows configuring multiple RADIUS authentication and accounting servers for redundancy and or load-balancing. RADIUS redundancy ensures high availability in subscriber authentication, authorization, and accounting services. You can configure multiple RADIUS servers, and when one goes down, another takes over.

The following algorithms are supported:

- **DIRECT (default):** The DIRECT algorithm is designed to primarily focus on **redundancy and fault tolerance**. In this mode, the RBFS subscriber daemon always attempts to send RADIUS requests to a single preferred server. If a response is received, the server continues to be used for subsequent requests. However, if the server becomes unresponsive or times out, the daemon fails over to the next server in the configured list. This process continues sequentially until a server responds or the list of servers is exhausted.
- **ROUND-ROBIN:** The ROUND-ROBIN algorithm implements **load-sharing** across all configured RADIUS servers. Each outgoing RADIUS request is sent to the next server in the list, circularly rotating through the servers. This allows the authentication and accounting load to be spread evenly across all available servers.

#### RADIUS NAS Port Identifier

RADIUS `NAS-Port-id` is used to identify the port of a network access router that the subscriber management feature uses to authenticate subscribers.

The RADIUS attribute NAS-Port-Id (87) is formatted such as the following:

```
<NAS-IDENTIFIER>#<IFP>#<OUTER-VLAN>#<INNER-VLAN>#<ACI>#<ARI>
```

The `Agent-Circuit-Id` (ACI) and `Agent-Remote-Id` (ARI) are replaced with an empty string (##) if not available.

### PPP over Ethernet (PPPoE)

PPP over Ethernet (PPPoE) is the common standard for internet access in the market.

#### PPPoE Session ID

The PPPoE session ID field is an unsigned 16-bit number as described in RFC2516. The values 0 and 65535 are reserved, with 0 used for PADI/PADO messages and 65535 reserved for future use. The session ID is unique within a broadcast domain (including IFP and VLANs) and for each subscriber device's MAC address. However, it is not unique per device or application instance. The session ID will change each time the session is reconnected.

#### PPPoE Service Name

When a subscriber device sends a PADI or PADR packet, the packet may include a `service name` field that specifies a particular service the client is requesting. Though, the system internally ignores the service name from the request, the name is still copied to the response packet sent back to the subscriber device.

If the request does not include any service name, the server's response (PADO or PADS) will include a default service name called `access` that ensures compatibility with certain clients, such as the Linux 'pppd'.

#### PPPoE AC-Cookie

This TAG is actually used to aid in protecting against denial-of-service attacks, but it is primarily used in RBFS to decide if a received PADR is a retry for an already answered (PADS send) one. The value itself is unpredictable and generated securely but it does not protect from reply attacks.

If a client receives this TAG in PADO, it MUST return the TAG unmodified in the following PADR. The TAG_VALUE is binary data of any value and length and is not interpreted by the Host.

The AC-Cookie is generated based on 8-bit salt followed by MD5 hash of salt, client MAC and dynamic PPPoE cookie secret.

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| SALT          | MD5                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The PPPoE cookie secret is randomly generated during the PPPoE daemon startup.

The AC-Cookie in the PADR creating the session is stored in the PPPoE PPP session object. For any received PADR it can be checked if there is a session on the same broadcast domain (IFP and VLAN's) and MAC with the same AC-Cookie. In this case, the PADS is just retried.

If the broadcast domain and MAC is equal but AC-Cookie is different, this PADR must be considered as a new request.

This allows to separate two different PPPoE sessions on the same VLAN from the same MAC as frequently used by some service providers.

When forwarding PPPoE packets, the TTL value is increased to '254' in the

outgoing IP packets.

#### PPPoE Session Limit

A customer line is typically represented by one (single-tagged) or two VLAN (double-tagged) on a physical interface with a limitation to one session, which is also called the 1:1 VLAN mode.

In some cases, the customer CPE will set up multiple PPPoE sessions on a single VLAN which requires MAC limitations greater than one but less or equal to the per VLAN limitation.

Therefore RBFS supports two different session limitations in the access interface configuration (Access Interface Configuration), one per VLAN (max-subscribers-per-vlan) and an additional per client MAC address (max-subscribers-per-mac) both set to 1 per default as required for 1:1 VLAN mode.

The limitation of sessions per client MAC address must be less or equal the sessions per VLAN and the default set to one for both limits.

#### PPPoE N:1 Multi-VRF Subscriber Support on Shared IFP/VLAN

RBFS supports PPPoE N:1 subscribers on the same IFP and VLAN, allowing them to be distributed across multiple routing instances (VRFs). Multiple VRFs can coexist on the same IFP and VLAN, ensuring that each individual subscriber session is associated with only one VRF. This enables logically separated services over shared access infrastructure, without interference between subscribers.

#### MTU Profiles

Bare metal switch hardware is typically limited in the number of supported MTU values. So RBFS has introduced the concept of MTU profiles with different types like `physical`, `ip`, `IPoE`, or `pppoe`. The last one is reserved for use with PPPoE sessions and applies to IPv4 and IPv6 traffic.

```
supervisor@switch: cfg> show config forwarding-options mtu-profile
{
  "rtbrick-config:mtu-profile": [
    {
      "mtu-profile-name": "IP-MTU-1500",
      "size": 1500,
      "type": "ip",
      "action": "redirect-to-cpu"
    },
    {
      "mtu-profile-name": "IP-MTU-9202",
      "size": 9202,
      "type": "ip",
      "action": "drop"
    },
    {
      "mtu-profile-name": "__default_pppoe__",
      "size": 1492,
```

```
      "type": "pppoe",
      "action": "redirect-to-cpu"
    }
  ]
}
```

> The physical access interface should be configured with an MTU profile large enough to serve all IPoE and PPPoE MTU profiles, including the additional overhead for PPPoE and VLAN headers. Further details about interface MTU profiles can be found in the *Interfaces Configuration Guide*.

The Q2a/Q2c platform supports up to 8 MTU profiles in total including the default PPPoE profile `default_pppoe`. The default profile cannot be deleted but overwritten to change size and action. For PPPoE type, you can create multiple profiles, but only one of the eight available MTU profiles is consumed for all configured PPPoE profiles. However, the action of all PPPoE MTU profiles must be the same.

The action could be either `drop` or `redirect-to-cpu`. The action `drop` silently discards all oversized packets. The action `redirect-to-cpu` punts oversized packets to the CPU where those could be either fragmented or dropped with ICMP response.

Fragmentation in software (cpu) must be explicitly enabled:

```
supervisor@switch: cfg> show config forwarding-options fragmentation
{
  "rtbrick-config:fragmentation": {
    "ipv4": {
      "state": "cpu"
    }
  }
}
```

**MTU Profile Support for PPPoE Subscribers**

The Point-to-Point Protocol (PPP) facilitates the negotiation of a Maximum Receive Unit (MRU) between endpoints, which is then applied as the Maximum Transmission Unit (MTU) on the opposite end of the PPP link. It is standard for each endpoint to independently negotiate its MRU, leading to potentially different MTU values for each direction of the PPP connection.

As an Internet Service Provider (ISP), control over the MTU/MRU sizes utilized for PPP over Ethernet (PPPoE) is not completely under your control. To address this, RBFS adopts a unique approach to managing PPPoE MTU profiles. Within RBFS, you can configure multiple PPPoE MTU profiles, however, they collectively count as a single entity against the eight MTU profile slots available on the Q2a/Q2c platform.

When a client initiates an MRU request via PPP Link Control Protocol (LCP) Configure-Request, RBFS employs this value to identify a matching MTU profile. This process involves sequentially comparing the requested MRU against an ordered list of MTU profiles until a profile with an MTU

size equal to or greater than the requested MRU is located. If an exact match is found, the corresponding profile is selected, the exact match counter is incremented, and the MRU is deemed acceptable.

If no exact match exists, the system selects the closest, yet smaller, MTU profile from the list. This selection increments the best match counter, and the MTU from this profile is proposed to the client through a PPP LCP Configure-Nak message. Should the client reject the proposed MRU after three attempts, RBFS applies the default MTU profile `default_pppoe` as fallback MTU profile. This protocol ensures that the client's requested MRU is honored, but the transmission is governed by the largest available PPPoE MTU profile, thereby maximizing client compatibility.

For example, if a client requests an MRU of 1482 bytes but only profiles for 1472 and 1492 bytes are configured, the system will offer 1472 bytes as the best match through a PPP LCP Configure-Nak. The client may accept this offer by submitting a new PPP LCP Configure-Request for 1472 bytes or persist with the initial request of 1482 bytes. After three iterations without agreement, the fallback profile is activated, accepting the client's MRU of 1482 bytes but applying the default MTU of 1492 bytes. This approach is justified as most Customer Premises Equipment (CPE) devices support TCP Maximum Segment Size (MSS) clamping based on the negotiated MRU, ensuring that at least TCP traffic adheres to the negotiated limit. Furthermore, it is not unusual for CPEs to accept packets exceeding the negotiated MRU, which is why a larger MTU profile is designated as the fallback.

The negotiated MRU and applied MTU can be verified with the following command for every single PPPoE session.

```
user@switch: op> show pppoe session 72339069014638648 detail
Subscriber-Id: 72339069014638648
    State: ESTABLISHED
    ...
    PPP LCP:
        ...
        MRU: 1492 Peer: 1492
        MTU: 1492 Profile: __default_pppoe__
    ...
```

The command `show pppoe mtu-profile` lists all PPPoE MTU profiles in increasing order, with two statistics about the exact and best match.

```
supervisor@switch: op> show pppoe mtu-profile
Profile                MTU      Exact Match      Best Match
PPPoE-MTU-1320         1320     0                0
PPPoE-MTU-1456         1456     0                0
PPPoE-MTU-1472         1472     0                0
__default_pppoe__      1492     0                0
```

The exact and best match counters can be used by operators to verify if the configured MTU profiles fit their environment or should be adopted.

As specified in RFC 2516, the Point-to-Point Protocol over Ethernet (PPPoE) requires a maximum

negotiated MTU/MRU of 1492 bytes. However, RFC 4638 introduces the PPP-Max-Payload Tag, which extends the maximum negotiable MTU/MRU beyond 1492 bytes. This enhancement aims to reduce fragmentation within emerging broadband networks.

To configure an MTU/MRU exceeding 1492 bytes, a PPPoE client must incorporate the PPP-Max-Payload Tag within the PADI and PADR packets. Should the PPP-Max-Payload Tag specify a value greater than 1492 bytes, and if either the PPP LCP MRU or the MTU outlined in the access-profile is also configured above 1492, the tag will be replicated into the PADO and PADS packets issued by the RBFS. Echoing the PPP-Max-Payload Tag to the client signales support for MTU/MRU beyond 1492 bytes.

The value indicated by the PPP-Max-Payload Tag establishes the maximum threshold for both MTU and MRU settings.

Therefore, the highest permissible MTU/MRU is determined by the smaller of the two values, the configured PPP LCP MTU/MRU or the PPP-Max-Payload Tag's value.

At least one of the following values must be set to a value of more than 1492 bytes:

```
set access access-profile pppoe-dual protocol ppp lcp mru 1500
set access access-profile pppoe-dual protocol ppp lcp mtu 1500
```

It is also required to set an MTU profile with the desired MTU size:

```
set forwarding-options mtu-profile pppoe-1500
set forwarding-options mtu-profile pppoe-1500 size 1500
set forwarding-options mtu-profile pppoe-1500 type pppoe
set forwarding-options mtu-profile pppoe-1500 action redirect-to-cpu
```

For L2TPv2 tunneled PPPoE sessions, the MTU is enforced by the LNS. It's usual behavior that the LNS renegotiates the MTU. So LAC may not know the actual MTU. This is the reason why RBFS does not apply an MTU profile for such sessions.

RBFS overwrites the selected MTU profile with `default_l2tp` for L2TPv2 sessions. This profile must be explicitly created, otherwise it is ignored. The action must be `drop` because ICMP or fragmentation is not supported for tunneled sessions.

```
supervisor@switch: cfg> show config forwarding-options mtu-profile __default_l2tp__
{
    "rtbrick-config:mtu-profile": [
      {
        "mtu-profile-name": "__default_l2tp__",
        "size": 1492,
        "type": "pppoe",
        "action": "drop"
      }
    ]
```

```
  }
```

This configuration allows to optionally enforce an MTU on LAC if needed.

Following an example MTU profile configuariton for PPPoE supporting all MTU values between the minimum of 1280 and maximum of 1500 in steps of 8 bytes.

```
supervisor@switch: cfg> show config forwarding-options mtu-profile
{
  "rtbrick-config:mtu-profile": [
    {
      "mtu-profile-name": "__default_pppoe__",
      "size": 1492,
      "type": "pppoe",
      "action": "redirect-to-cpu"
    },
    {
      "mtu-profile-name": "PPPoE-MTU-1280",
      "size": 1280,
      "type": "pppoe",
      "action": "redirect-to-cpu"
    },
    {
      "mtu-profile-name": "PPPoE-MTU-1288",
      "size": 1288,
      "type": "pppoe",
      "action": "redirect-to-cpu"
    },
    {
      "mtu-profile-name": "PPPoE-MTU-1296",
      "size": 1296,
      "type": "pppoe",
      "action": "redirect-to-cpu"
    },
    {
      "mtu-profile-name": "PPPoE-MTU-1304",
      "size": 1304,
      "type": "pppoe",
      "action": "redirect-to-cpu"
    },
    {
      "mtu-profile-name": "PPPoE-MTU-1312",
      "size": 1312,
      "type": "pppoe",
      "action": "redirect-to-cpu"
    },
    {
      "mtu-profile-name": "PPPoE-MTU-1320",
      "size": 1320,
      "type": "pppoe",
```

```
      "action": "redirect-to-cpu"
    },
    {
      "mtu-profile-name": "PPPoE-MTU-1328",
      "size": 1328,
      "type": "pppoe",
      "action": "redirect-to-cpu"
    },
    {
      "mtu-profile-name": "PPPoE-MTU-1336",
      "size": 1336,
      "type": "pppoe",
      "action": "redirect-to-cpu"
    },
    {
      "mtu-profile-name": "PPPoE-MTU-1344",
      "size": 1344,
      "type": "pppoe",
      "action": "redirect-to-cpu"
    },
    {
      "mtu-profile-name": "PPPoE-MTU-1352",
      "size": 1352,
      "type": "pppoe",
      "action": "redirect-to-cpu"
    },
    {
      "mtu-profile-name": "PPPoE-MTU-1360",
      "size": 1360,
      "type": "pppoe",
      "action": "redirect-to-cpu"
    },
    {
      "mtu-profile-name": "PPPoE-MTU-1368",
      "size": 1368,
      "type": "pppoe",
      "action": "redirect-to-cpu"
    },
    {
      "mtu-profile-name": "PPPoE-MTU-1376",
      "size": 1376,
      "type": "pppoe",
      "action": "redirect-to-cpu"
    },
    {
      "mtu-profile-name": "PPPoE-MTU-1384",
      "size": 1384,
      "type": "pppoe",
      "action": "redirect-to-cpu"
    },
    {
```

```
      "mtu-profile-name": "PPPoE-MTU-1392",
      "size": 1392,
      "type": "pppoe",
      "action": "redirect-to-cpu"
    },
    {
      "mtu-profile-name": "PPPoE-MTU-1400",
      "size": 1400,
      "type": "pppoe",
      "action": "redirect-to-cpu"
    },
    {
      "mtu-profile-name": "PPPoE-MTU-1408",
      "size": 1408,
      "type": "pppoe",
      "action": "redirect-to-cpu"
    },
    {
      "mtu-profile-name": "PPPoE-MTU-1416",
      "size": 1416,
      "type": "pppoe",
      "action": "redirect-to-cpu"
    },
    {
      "mtu-profile-name": "PPPoE-MTU-1424",
      "size": 1424,
      "type": "pppoe",
      "action": "redirect-to-cpu"
    },
    {
      "mtu-profile-name": "PPPoE-MTU-1432",
      "size": 1432,
      "type": "pppoe",
      "action": "redirect-to-cpu"
    },
    {
      "mtu-profile-name": "PPPoE-MTU-1440",
      "size": 1440,
      "type": "pppoe",
      "action": "redirect-to-cpu"
    },
    {
      "mtu-profile-name": "PPPoE-MTU-1448",
      "size": 1448,
      "type": "pppoe",
      "action": "redirect-to-cpu"
    },
    {
      "mtu-profile-name": "PPPoE-MTU-1456",
      "size": 1456,
      "type": "pppoe",
```

```
      "action": "redirect-to-cpu"
    },
    {
      "mtu-profile-name": "PPPoE-MTU-1464",
      "size": 1464,
      "type": "pppoe",
      "action": "redirect-to-cpu"
    },
    {
      "mtu-profile-name": "PPPoE-MTU-1472",
      "size": 1472,
      "type": "pppoe",
      "action": "redirect-to-cpu"
    },
    {
      "mtu-profile-name": "PPPoE-MTU-1480",
      "size": 1480,
      "type": "pppoe",
      "action": "redirect-to-cpu"
    },
    {
      "mtu-profile-name": "PPPoE-MTU-1488",
      "size": 1488,
      "type": "pppoe",
      "action": "redirect-to-cpu"
    },
    {
      "mtu-profile-name": "PPPoE-MTU-1490",
      "size": 1490,
      "type": "pppoe",
      "action": "redirect-to-cpu"
    },
    {
      "mtu-profile-name": "PPPoE-MTU-1500",
      "size": 1500,
      "type": "pppoe",
      "action": "redirect-to-cpu"
    }
  ]
}
```

**MTU Profile Support for IPoE Subscribers**

Unlike the PPPoE protocol, which can negotiate MTU dynamically, IPoE relies on a fixed MTU setting. The MTU value is predefined and remains constant for all IPoE subscribers using the configured access interface.

For IPoE subscribers, MTU is statically configured and is applied at the access interface level.

Configuration commands for the MTU profile for IPoE subscribers.

```
set access access-profile ipoe
set access access-profile ipoe instance default
set access access-profile ipoe protocol dhcp enable true
set access access-profile ipoe protocol dhcp lease-time 3600
set access access-profile ipoe protocol dhcpv6 enable true
set access access-profile ipoe protocol dhcpv6 lifetime 3600
set access access-profile ipoe protocol ipoe ip-mtu-profile ipoe_mtu_1500
```

It is also required to set an MTU profile with the desired MTU size:

```
set forwarding-options mtu-profile ipoe_mtu_1500
set forwarding-options mtu-profile ipoe_mtu_1500 size 1500
set forwarding-options mtu-profile ipoe_mtu_1500 type ip
set forwarding-options mtu-profile ipoe_mtu_1500 action drop
```

Following is an example of access profile IPoE:

```
supervisor@rtbrick.net: cfg> show config access access-profile ipoe
{
  "rtbrick-config:access-profile": [
    {
      "profile-name": "ipoe",
      "instance": "default",
      "protocol": {
        "dhcp": {
          "enable": "true",
          "lease-time": 3600
        },
        "dhcpv6": {
          "enable": "true",
          "lifetime": 3600
        },
        "ipoe": {
          "ip-mtu-profile": "ipoe_mtu_1500"
        }
      },
      "address-family": {
        "ipv4": {
          "enable": "true",
          "pool-name": "poolv4"
        },
        "ipv6": {
          "enable": "true",
          "pool-name": "poolv6",
          "prefix-delegation-pool-name": "poolv6pd"
        }
      }
    }
```

```
  ]
}
```

Following an example MTU profile configuariton for IPoE:

```
supervisor@DT-STD-23-2401>bm13-tst.fsn.rtbrick.net: cfg> show config forwarding-
options mtu-profile
{
  "rtbrick-config:mtu-profile": [
    {
      "mtu-profile-name": "ipoe_mtu_1500",
      "size": 1500,
      "type": "ip",
      "action": "drop"
    }
  ]
}
```

#### PPPoE VLAN Profiles

This chapter describes the VLAN profile feature. If enabled for the access interface, then incoming sessions (e.g. PPPoE PADI/PADR) are not honored unless matching vlan-profile is found.

The VLAN profiles must be added to the table `global.vlan.profile` owned by PPPoE daemon. All entries in this table are ephemeral and therefore lost after reboot or PPPoE daemon restart.

**Example:**

```
{
    "table": {
        "table_name": "global.vlan.profile"
    },
    "objects": [
        {
            "attribute": {
                "ifp_name": "ifp-0/1/2",
                "outer_vlan_min": 128,
                "outer_vlan_max": 128,
                "inner_vlan_min": 1,
                "inner_vlan_max": 4095,
                "access_profile_name": "access-profile-vlan"
            }
        }
    ]
}
```

#### PPPoE Dual-Stack IPv4/IPv6 with DHCPv6

The whole IPv6 control plane of a PPPoE session like ICMPv6 router-solicitation (RS), ICMPv6 router-advertisement (RA) and DHCPv6 is handled in the PPPoE daemon (pppoed).

The PPPoE daemon handles received router solicitations by responding with router advertisements and is sending frequent router advertisements based on configured intervals.

The other-config flag in the router-advertisement is automatically set if DHCPv6 is enabled for this particular subscriber. This flag signals that there is more information available via DHCPv6.

DHCPv6 over PPPoE is different to DHCPv6 over Ethernet because of the special characteristics of point-to-point protocols. DHCPv6 over PPPoE is supporting delegated IPv6 prefixes (IA_PD) and DNS options only. Unsupported IA options (IA_NA and IA_TA) or options that can be served will be rejected with status code options as defined per RFC.

The delegated IPv6 prefix served by DHCPv6 will be assigned to the subscriber via RADIUS or local pool regardless of the protocols negotiated with the client. DHCPv6 was primarily designed for use in Ethernet networks. The fact that Ethernet is connectionless requires that DHCPv6 servers must manage releases for the clients and free them automatically if a lease expires. Such extensive release management is not needed for connection-oriented protocols like PPPoE where addresses are assigned to the PPPoE session. This fact allows to implementing DHCPv6 nearly stateless on the server side by just tracking if an assigned prefix is assigned or released. This is tracked in the attribute `ipv6pd_negotiated` of the PPPoED/SubscriberD (global.ppp.1.subscriber.result) result object and copied to the actual subscriber object (`local.access.subscriber`). As this use case is covered by PPPoE, there is no lease expiry implemented.

The delegated-prefix is added to the subscriber-ifl only if negotiated and removed if not negotiated. The presence of delegated prefix in the subscriber-ifl is used by IFMD to add or remove the forwarding entry.

If DHCPv6 is enabled but no delegated prefix provided, only DNS is served in response if available.

#### PPPoE DHCPv6 Server DUID

The DHCPv6 server identifier DUID is generated based on IP6CP negotiated interface-identifier as shown below:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     DUID-Type 3 (DUID-LL)     |    hardware type 27 (EUI64)   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       interface-identifier                    |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

#### Support for Preventing Simultaneous PPPoE and IPoE Subscriber Sessions on the Same VLAN

RBFS can prevent simultaneous IPoE and PPPoE subscriber sessions on the same VLAN. While the

system allows either access type (IPoE or PPPoE) to be dynamically accepted on a shared VLAN, it does not support both IPoE and PPPoE subscriber sessions active at the same time. The subscriber daemon can detect existing subscriber access type on an IFP VLAN and reject any new subscriber request that conflicts with the currently active access type.

**Layer Two Tunneling Protocol (L2TPv2)**

This chapter describes the RtBrick Layer Two Tunneling Protocol (L2TPv2) implementation. This document describes also the corresponding configuration and operations commands for PPPoE access services with PPP tunneling using the Layer Two Tunneling Protocol version 2 (L2TPv2) on RtBRick FullStack (RBFS).

Typically, a user obtains a Layer 2 (L2) point-to-point connection to a Broadband Network Gateway (BNG) using the PPPoE protocol as described in RFC 2516 and runs PPP over that connection. In the most common case, the L2 termination point and PPP session endpoint reside on the same physical device. Tunneling protocols, such as L2TPv2 provide a dynamic mechanism for extending PPP by allowing the L2 and PPP endpoints reside on different devices that are interconnected by an IP network. This separation allows the actual processing of PPP packets to be divorced from the termination of the L2 circuit. The L2TP access concentrator (LAC) physically terminates the L2 connection and tunnels the PPP packets across an IP network to the L2TP network server (LNS). The LNS then terminates the logical PPP connection.

RFC and draft compliance are partial except as specified.

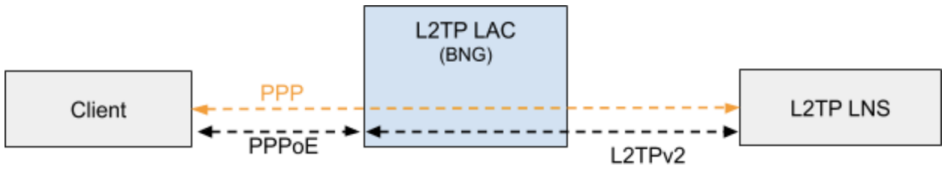



*Figure 61. L2TP PPPoE*

To establish a PPPoE session via L2TP, the tunnel-type must be configured as L2TP. This configuration can be achieved either for local users or by utilizing the corresponding tunnel-type attribute through RADIUS.

```
# Local User
cfg> set access user-profile local@l2tp tunnel-type L2TP

# FreeRADIUS
"radius@l2tp" Cleartext-Password := "test"
    Service-Type = Framed-User,
    Framed-Protocol = PPP,
    Tunnel-Type:0 = L2TP
```

Defining an L2TP configuration profile is essential, which can be referenced through an access-profile or by employing the appropriate RADIUS VSA. The actual tunnels may either be defined locally via an L2TP pool configuration or set up dynamically through RADIUS.

#### L2TP LAC

The L2TP Access Concentrator (LAC) is a node that acts as one side of an L2TP tunnel endpoint, and is a peer to the L2TP Network Server L2TP LNS. The LAC sits between a LNS and a remote system, and forwards packets to and from each.

#### L2TP LNS

The L2TP Network Server (LNS) is a node that acts as one side of an L2TP tunnel endpoint and is a peer to the L2TP Access Concentrator L2TP LAC. The LNS is the logical termination point of a PPP session that is being tunneled from the remote system by the LAC.

> The LNS role is currently not supported!

#### L2TP Tunnel Selection

Each new session creates a session request object (local.l2tp.session.request) to track the tunnel selection progress, the currently selected ones, and which are already tried. This object is automatically deleted if the session setup is successful.

All tunnels in state DEAD are skipped in the tunnel selection but considered at the end if no other tunnels are available. Tunnels with a session limit reached are not considered for further sessions. To select a tunnel, the L2TP daemon first generates list of preferred tunnels based on tunnel preference, where the lowest value has the highest priority. The configured L2TP tunnel selection algorithm decides how to select a tunnel out of the remaining tunnels with the same preference. The RANDOM algorithm selects the tunnel randomly whereas BALANCED selects the least filled tunnel based on the number of sessions.

Following the L2TP tunnel pool order/priority in case, there are multiple pools available for a single subscriber:

1. RADIUS defined tunnel (RFC2866)
2. RADIUS VSA (RtBrick-L2TP-Pool) or local user profile
3. L2TP configuration profile

#### L2TP Control Channel

The control channel is responsible for the orderly passing of control messages between the tunnel endpoints and acts as a transport layer for reliable delivery of control messages and tunnel keep alive services for the tunnel.

Each L2TP tunnel is split into the actual tunnel object with all the information exchanged during tunnel establishment plus the FSM state and a separate control channel with the sequence numbers, window size, and thresholds changed with every sent and received packet.

RBFS sent a ZLB ACK only if there are no further messages waiting in queue for that peer, as well as to acknowledge multiple packets at once.

The HELLO keep-alive messages are also part of the control channel and only send if there is no other message sent if the queue is empty and no other message send during the hello interval.

#### Support N:1 VLAN (Shared VLAN) Model for L2TP Subscribers

RBFS supports for L2TP subscribers using the N:1 VLAN model, also known as the shared VLAN model. This feature is enabled, by default.

#### L2TP Access Line Information (RFC5515)

#### Connect-Speed-Update-Notification (CSUN)

The Connect-Speed-Update-Notification (CSUN) is an L2TP control message sent by the LAC to the LNS to provide transmit and receive connection speed updates for one or more sessions.

> This implementation will send one CSUN request per session!

CSUN requests are disabled per default and can be enabled in the L2TP profile L2TP Profile Configuration.

CSUN messages are defined in RFC5515, which is not widely supported. Therefore those messages are marked as not mandatory in RBFS to allow interwork with LNS servers not supporting RFC5515.

**RFC2661:**

```
The Mandatory (M) bit within the Message Type AVP has special
meaning. Rather than an indication of whether the AVP itself
should be ignored if not recognized, it is an indication as to
whether the control message itself should be ignored. Thus, if the
M-bit is set within the Message Type AVP and the Message Type is
unknown to the implementation, the tunnel MUST be cleared.  If the
M-bit is not set, then the implementation may ignore an unknown
message type.
```

> RFC and draft compliance are partial except as specified.

#### Connect-Speed-Update-Request (CSURQ)

The Connect-Speed-Update-Request (CSURQ) is an L2TP control message sent by the LNS to the LAC to request the current transmission and receive connection speed for one or more sessions.

> Sending or responding to CSURQ requests is currently not supported!

#### Access Line Information L2TP Attribute Value Pair Extensions

The corresponding access line information for a subscriber is included in the ICRQ message as defined in RFC5515.

#### Connect Speed Values

The default value for TX and RX Connect Speed is set to 1000000000 (1G) which is replaced by the actual data rate upstream/downstream of the corresponding access line information object or

directly set using the RADIUS attributes RtBrick-L2TP-Tx-Connect-Speed (42) and RtBrick-L2TP-Rx-Connect-Speed (43).

**IPoE**

IP-over-Ethernet (IPoE) is a popular alternative to PPPoE based access using DHCP for IPv4 and DHCPv6 for IPv6 where both protocols are handled in the IPoE daemon (ipoed).

IPoE subscribers are identified by Agent-Remote-Id, Agent-Circuit-Id, interface, VLAN or MAC address.

The dynamic creation of IPoE subscribers is triggered by the first DHCPv4 discovery or DHCPv6 solicit request received. Any response is postponed until the subscriber is successfully authenticated using known authentication methods like none, local or RADIUS. For DHCP mode `server` all addresses are assigned during authentication to the subscriber and used by DHCPv4 and DHCPv6 to handle client requests. For the DHCP relay mode, all IP addresses are allocated by an external DHCP server. Forwarding of requests to the configured servers is also postponed until the subscriber is authenticated.

IPoE subscribers will be terminated automatically if all protocol bindings have been expired or failed.

**IPoE VLAN Support**

RBFS supports both VLAN models, 1:1 and N:1 for IPoE subscribers.

**1:1 VLAN (VLAN-per-subscriber) Model**

In the 1:1 VLAN model, there is a unique dedicated customer VLAN for a subscriber, that is one VLAN per subscriber. This model establishes a unique path between each subscriber interface and the router for data transmission by providing traffic separation for every subscriber.

1:1 model operations are relatively simple as it provides one-to-one mapping of specific VLANs to specific subscribers. New services can be added easily without affecting other subscribers and services with this model.

The following diagram shows a dedicated customer VLAN for a subscriber, that is, one VLAN per subscriber.

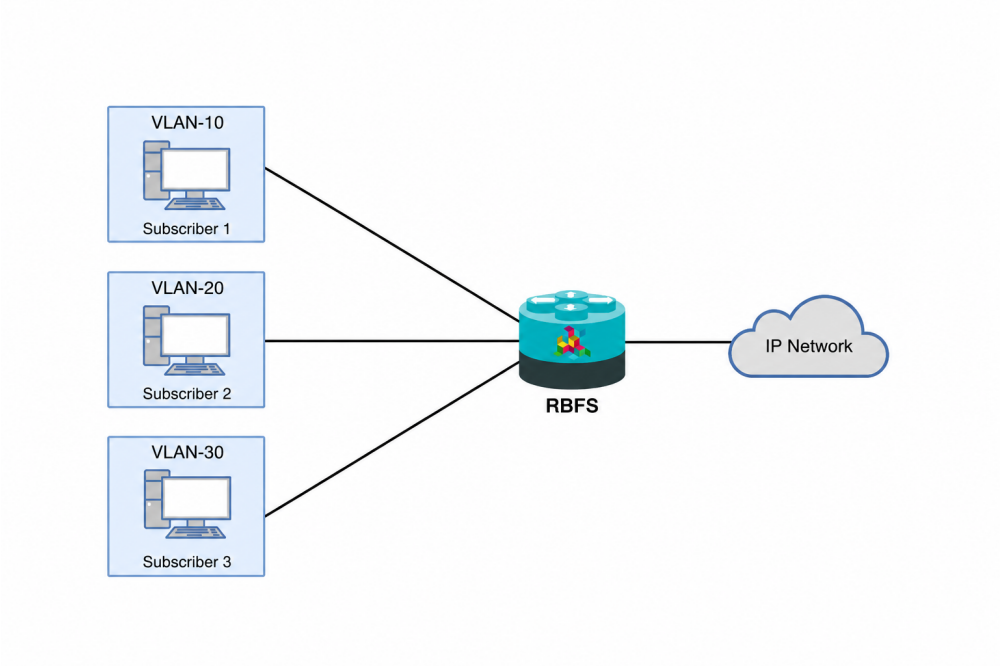



*Figure 62. 1:1 VLAN (VLAN-per-subscriber)*

#### N:1 VLAN (Shared VLAN) Model

The shared VLAN model provides many-to-one (N:1) subscriber-to-service connectivity. This model provides a single VLAN to many subscribers. Unlike in the 1:1 model, in which the VLAN is dedicated to a customer, N:1 provides a shared VLAN to many subscribers, and this VLAN carries all types of service (i.e., data, voice, video, etc.). One disadvantage of shared VLAN is the lack of logical isolation between user sessions at the VLAN level.

The following diagram shows a single VLAN that is connected to many subscribers.

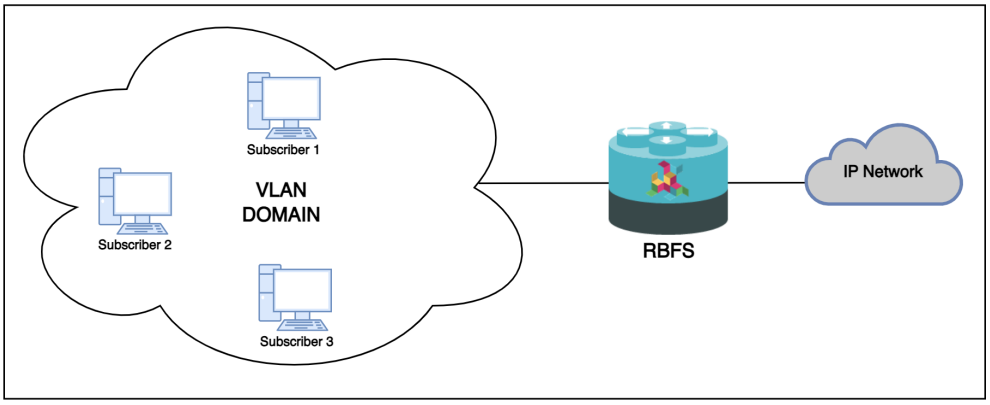



*Figure 63. N:1 VLAN (Shared VLAN-per-service)*

### Limitations for IPoE N:1 VLAN

Access services such as http-redirect and subscriber-acl are currently not supported for IPoE N:1 subscribers.

#### IPoE Session Limit

A customer line is typically represented by one (single-tagged) or two VLAN (double-tagged) on a physical interface with a limitation to one subscriber, which is also called the 1:1 VLAN mode.

IPoE subscribers are implicitly limited to max one per MAC address, as client MAC address is used as part of the key to identify subscribers.

#### IPoE Username

For each IPoE subscriber, a username is generated automatically using the client's MAC address followed by `@ipoe`.

Example: `fe:08:e8:ea:1d:32@ipoe`

#### IPoE DHCPv6 Server DUID

The generated DHCPv6 server identifier DUID is from type 3 (DUID-LL) with hardware type 27 (EUI64) and IFP MAC address to derive the EUI64 interface identifier.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     DUID-Type 3 (DUID-LL)     |    hardware type 27 (EUI64)   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      interface-identifier                     |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

#### IPoE DHCP Relay

When IPoE is configured on relay mode, the system relies on an external DHCP server for IP address allocation and client configuration functions. The DHCP server contains a pool of IP addresses, and from that pool, it allocates addresses to subscribers. RBFS acts as a DHCP relay during its interaction with the DHCP server. The feature allows multiple RBFS instances to use a single centralized DHCP server for their IP allocation.

For information on how to configure IPoE in relay mode, see the sections DHCPv4 and DHCPv6.

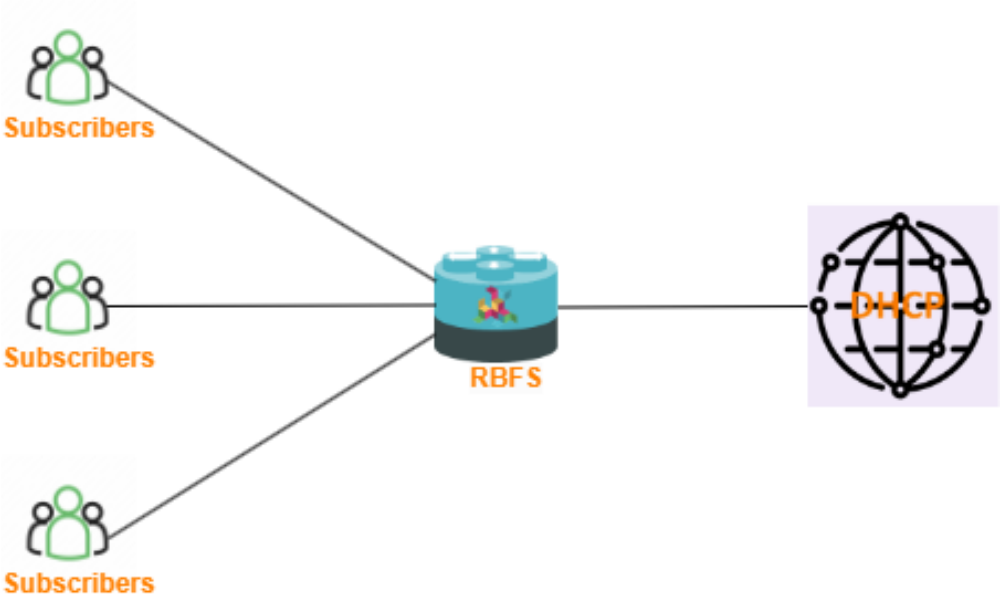



*Figure 64. IPoE DHCP Relay*

For DHCPv4 relay, the gateway IFL's primary address is used as GIADDR. This address may also serve as the default gateway for clients, although it is not necessarily so. Since we assume proxy ARP is enabled for all addresses, customers can assign addresses in blocks and choose a gateway from within that block. This setup works even if the chosen IP address is not configured on our BNG, as proxy ARP resolves it.

For DHCPv6, only the primary global address from the gateway IFL is used as the link address towards the server. If this is not set, the link-local address is used, which is always employed as the gateway for the client by ICMPv6 RA.

For DHCPv6, RBFS constructs the Interface-Id as shown below.

```
<ifp>#<outer-vlan>#<innter-vlan>#<subscriber-id>
```

It is important to note that RBFS sets the Interface-Id only on its own relay header, leaving any nested headers (such as those added by LDRA at the OLT) unchanged. This allows DHCPv6 servers to match individually on those nested headers.

#### DHCP Asymmetric Lease Support

RBFS supports DHCP Asymmetric Leases for both DHCPv4 and DHCPv6 in relay mode. This feature offers two independent, yet complementary, capabilities designed to optimize IPoE subscriber management and offer better control over lease timers.

### Custom Timer and Option Overrides

DHCP Asymmetric Lease support explicit renewal and rebind timer settings for both DHCPv4 and DHCPv6 across server and relay operational modes.

Timer Manipulation: In relay mode, RBFS dynamically overwrites these timers in messages sent to the client without altering the local binding timers. Furthermore, other critical timers, such as DHCPv4 lease time, or DHCPv6 lifetime and preferred-lifetime, can be independently overwritten

while preserving the original lease time issued by the server for the local binding.

DNS Overrides: If DNS servers are specified via local configuration or RADIUS, they will automatically override any DNS servers provided by the upstream DHCP server.

**Asymmetric Lease Operation**

When Asymmetric Lease is enabled, the RBFS acts like local proxy for client renewals to reduce traffic to the upstream server:

As long as the actual local lease timer is less than 50% expired, the BNG will intercept the client's renewal request and reply directly based on the last lease acknowledgement packet received from the server.

Once the lease timer is 50% or more expired, the BNG will forward the renewal request to the upstream DHCP server.

#### IPoE DHCPv4 Options Supported

The following table provides an overview of all options supported by RBFS.

In relay mode, unsupported options are still forwarded along with the supported ones.

*DHCP Options*

| Options | Sub-options | Description | Comment |
|---|---|---|---|
| 0 | - | Pad | - |
| 1 | - | Subnet Mask | - |
| 3 | - | Router | - |
| 6 | - | Domain Server | - |
| 50 | - | Address Request | - |
| 51 | - | Address Time | - |
| 53 | - | DHCP message type | - |
| 54 | - | DHCP Server ID | - |
| 55 | - | Parameter list | - |
| 56 | - | DHCP message | - |
| 61 | - | Client ID | - |
| 82 | - | Relay agent information | - |
| 82 | 1 | Agent circuit ID | - |
| 82 | 2 | Agent Remote ID | - |
| 82 | 9 | Vendor Specific | - |
| 255 | - | End | - |

The following table provides vendor-specific relay options (DHCP Option 82, sub-option 9). Access line attributes within Broadband Forum (3561) options are recognized by RBFS as access line attributes and are also forwarded to RADIUS.

*DHCP Vendor Options*

| Vendor Options | Sub-options | Description | Comment |
|---|---|---|---|
| 3561 | 129 | Actual Data Rate Upstream | - |
| 3561 | 130 | Actual Data Rate Downstream | - |
| 3561 | 131 | Minimum Data Rate Upstream | - |
| 3561 | 132 | Minimum Data Rate Downstream | - |
| 3561 | 133 | Attainable DataRate Upstream | - |
| 3561 | 134 | Attainable DataRate Downstream | - |
| 3561 | 135 | Maximum Data Rate Upstream | - |
| 3561 | 136 | Maximum Data Rate Downstream | - |
| 3561 | 137 | Minimum Data Rate Upstream in low power state | - |
| 3561 | 138 | Minimum Data Rate Downstream in low power state | - |
| 3561 | 139 | Maximum Interleaving Delay Upstream | - |
| 3561 | 140 | Actual Interleaving Delay Upstream | - |
| 3561 | 141 | Maximum Interleaving Delay Downstream | - |
| 3561 | 142 | Actual Interleaving Delay Downstream | - |
| 3561 | 144 | Data Link Encapsulation | - |
| 3561 | 145 | DSL Type | - |
| 3561 | 146 | PON-Access-Type | Backward compatibility with LIHAWI draft version 00. |
| 3561 | 147 | ONT/ONU-Average-Data-Rate-Downstream | Backward compatibility with LIHAWI draft version 00. |
| 3561 | 148 | ONT/ONU-Peak-Data-Rate-Downstream | Backward compatibility with LIHAWI draft version 00. |
| 3561 | 149 | ONT/ONU-Maximum-Data-Rate-Upstream | Backward compatibility with LIHAWI draft version 00. |
| 3561 | 150 | ONT/ONU-Assured-Data-Rate-Upstream | Backward compatibility with LIHAWI draft version 00. |

| Vendor Options | Sub-options | Description | Comment |
|---|---|---|---|
| 3561 | 151 | PON-Access-Type / PON-Tree-Maximum-Data-Rate-Upstream | If value is > 999, RBFS interprets as PON-Tree-Maximum-Data-Rate-Upstream as defined by LIHAWI version 00, otherwise as PON-Access-Type. |
| 3561 | 152 | PON-Tree-Maximum-Data-Rate-Downstream | Backward compatibility with LIHAWI draft version 00. |
| 3561 | 155 | Expected Throughput (ETR) Upstream | - |
| 3561 | 156 | Expected Throughput (ETR) Downstream | - |
| 3561 | 157 | Attainable Expected Throughput (ATTETR) Upstream | - |
| 3561 | 158 | Attainable Expected Throughput (ATTETR) Downstream | - |
| 3561 | 159 | Gamma Data Rate (GDR) Upstream | - |
| 3561 | 160 | Gamma Data Rate (GDR) Downstream | - |
| 3561 | 161 | Attainable Gamma Data Rate (ATTGDR) Upstream | - |
| 3561 | 162 | Attainable Gamma Data Rate (ATTGDR) Downstream | - |
| 3561 | 176 | ONT/ONU-Average-Data-Rate-Downstream | - |
| 3561 | 177 | ONT/ONU-Peak-Data-Rate-Downstream | - |
| 3561 | 178 | ONT/ONU-Maximum-Data-Rate-Upstream | - |
| 3561 | 179 | ONT/ONU-Assured-Data-Rate-Upstream | - |
| 3561 | 180 | PON-Tree-Maximum-Data-Rate-Upstream | - |
| 3561 | 181 | PON-Tree-Maximum-Data-Rate-Downstream | - |

**IPoE DHCPv6 Options Supported**

The following table provides the list of all the IPoE DHCPv6 options supported by RBFS. In relay mode, unsupported options are still forwarded along with the supported ones.

*DHCP Options*

| Options | Sub-options | Description | Comment |
|---|---|---|---|
| 1 | - | OPTION_CLIENTID | - |
| 2 | - | OPTION_SERVERID | - |
| 3 | - | OPTION_IA_NA | - |

| Options | Sub-options | Description | Comment |
|---|---|---|---|
| 6 | - | OPTION_ORO | - |
| 8 | - | OPTION_ELAPSED_TIME | - |
| 9 | - | OPTION_RELAY_MSG | - |
| 13 | - | OPTION_STATUS_CODE | - |
| 14 | - | OPTION_RAPID_COMMIT | - |
| 17 | - | OPTION_VENDOR_OPTS | - |
| 18 | - | OPTION_INTERFACE_ID | - |
| 23 | - | OPTION_DNS_SERVERS | - |
| 25 | - | OPTION_IA_PD | - |
| 26 | 1 | OPTION_IAPREFIX | - |
| 37 | 2 | OPTION_REMOTE_ID | - |

The list below outlines vendor-specific options (DHCP Option 17). Access line attributes within Broadband Forum (3561) options are recognized by RBFS as access line attributes and are also forwarded to RADIUS.

*DHCPv6 Vendor Options*

| Vendor Options | Sub-options | Description | Comment |
|---|---|---|---|
| 3561 | 129 | Actual Data Rate Upstream | - |
| 3561 | 130 | Actual Data Rate Downstream | - |
| 3561 | 131 | Minimum Data Rate Upstream | - |
| 3561 | 132 | Minimum Data Rate Downstream | - |
| 3561 | 133 | Attainable DataRate Upstream | - |
| 3561 | 134 | Attainable DataRate Downstream | - |
| 3561 | 135 | Maximum Data Rate Upstream | - |
| 3561 | 136 | Maximum Data Rate Downstream | - |
| 3561 | 137 | Minimum Data Rate Upstream in low power state | - |
| 3561 | 138 | Minimum Data Rate Downstream in low power state | - |
| 3561 | 139 | Maximum Interleaving Delay Upstream | - |
| 3561 | 140 | Actual Interleaving Delay Upstream | - |
| 3561 | 141 | Maximum Interleaving Delay Downstream | - |
| 3561 | 142 | Actual Interleaving Delay Downstream | - |

| Vendor Options | Sub-options | Description | Comment |
|---|---|---|---|
| 3561 | 144 | Data Link Encapsulation | - |
| 3561 | 145 | DSL Type | - |
| 3561 | 146 | PON-Access-Type | Backward compatibility with LIHAWI draft version 00. |
| 3561 | 147 | ONT/ONU-Average-Data-Rate-Downstream | Backward compatibility with LIHAWI draft version 00. |
| 3561 | 148 | ONT/ONU-Peak-Data-Rate-Downstream | Backward compatibility with LIHAWI draft version 00. |
| 3561 | 149 | ONT/ONU-Maximum-Data-Rate-Upstream | Backward compatibility with LIHAWI draft version 00. |
| 3561 | 150 | ONT/ONU-Assured-Data-Rate-Upstream | Backward compatibility with LIHAWI draft version 00. |
| 3561 | 151 | PON-Access-Type / PON-Tree-Maximum-Data-Rate-Upstream | If value is > 999, RBFS interprets as PON-Tree-Maximum-Data-Rate-Upstream as defined by LIHAWI version 00, otherwise as PON-Access-Type. |
| 3561 | 152 | PON-Tree-Maximum-Data-Rate-Downstream | Backward compatibility with LIHAWI draft version 00. |
| 3561 | 155 | Expected Throughput (ETR) Upstream | - |
| 3561 | 156 | Expected Throughput (ETR) Downstream | - |
| 3561 | 157 | Attainable Expected Throughput (ATTETR) Upstream | - |
| 3561 | 158 | Attainable Expected Throughput (ATTETR) Downstream | - |
| 3561 | 159 | Gamma Data Rate (GDR) Upstream | - |
| 3561 | 160 | Gamma Data Rate (GDR) Downstream | - |
| 3561 | 161 | Attainable Gamma Data Rate (ATTGDR) Upstream | - |
| 3561 | 162 | Attainable Gamma Data Rate (ATTGDR) Downstream | - |
| 3561 | 176 | ONT/ONU-Average-Data-Rate-Downstream | - |
| 3561 | 177 | ONT/ONU-Peak-Data-Rate-Downstream | - |
| 3561 | 178 | ONT/ONU-Maximum-Data-Rate-Upstream | - |
| 3561 | 179 | ONT/ONU-Assured-Data-Rate-Upstream | - |
| 3561 | 180 | PON-Tree-Maximum-Data-Rate-Upstream | - |

| Vendor Options | Sub-options | Description | Comment |
|---|---|---|---|
| 3561 | 181 | PON-Tree-Maximum-Data-Rate-Downstream | - |

**Supported Platforms**

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

## 5.1.2. Subscriber Management Configuration Overview

RBFS Subscriber Management configuration involves setting up various profiles and parameters that control how subscribers interact with network services. This includes authentication, service access, protocol handling, and other functions organized under a structured configuration hierarchy. The profiles determine how subscribers authenticate, access network services, and use various protocols. The configuration starts with mandatory settings such as interface, access, and AAA profiles, followed by optional configurations for more features such as RADIUS, L2TP, and service profiles. Each profile plays a crucial role in ensuring efficient and secure subscriber management.

**Configuration Hierarchy**

The configuration of physical interfaces (IFP) and their associated VLANs is managed through a set of profiles that define parameters for various functions. These include authentication settings with AAA, service management for protocols such as IGMP and MLD, and access protocols such as PPPoE.

The following image illustrates how the subscriber management configuration and profile system are organized.

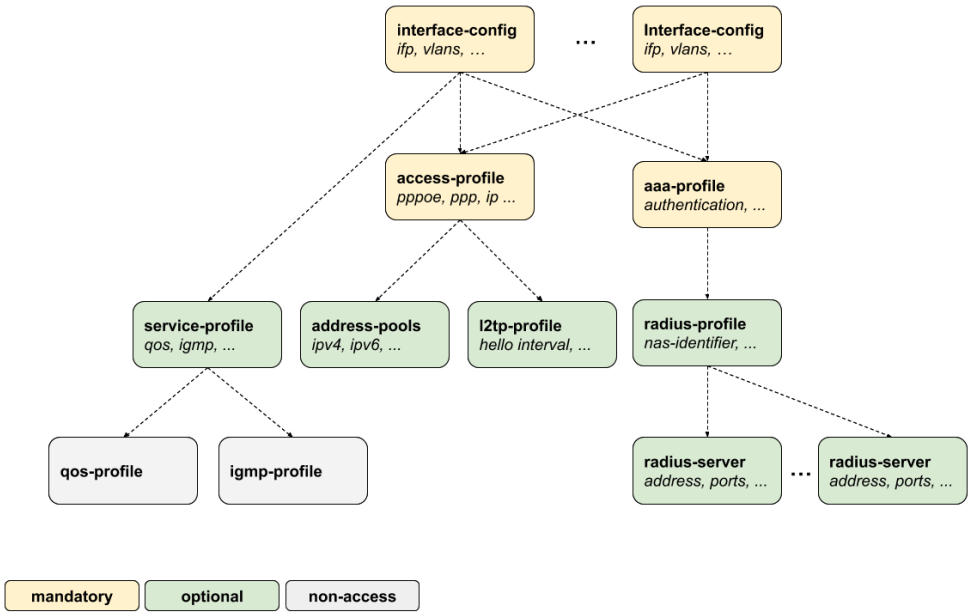



*Figure 65. Configuration and Profiles*

All subscriber management configurations and profiles are managed under the top-level hierarchy `access`. This hierarchy acts as the central point for defining and managing various access protocols, subscriber management profiles, and authentication settings that are crucial for network operations.

The following is the high-level command syntax and the table provides command options and descriptions available for Subscriber Management configurations.

**Syntax:**

**set access aaa-profile** <profile-name> <option>
**set access access-profile** <profile-name> <option>
**set access chassis-id** [0 - 15]
**set access dhcp-relay** <interface> <option>
**set access dhcpv6-server** <server-name> <option>
**set access dhcpv6-server** <server-name>
**set access interface** <option>
**set access l2tp-pool** <pool-name> <client-name> <server-name> <option>
**set access l2tp-profile** <profile-name> <option>
**set access pool** <pool-name> <option>
**set access radius-profile** <profile-name> <option>
**set access radius-server** <server-name> <option>
**set access service-profile** <profile-name> <option>
**set access terminate-history-timeout** [0 - 720]
**set access user-profile** <user-name> <option>

| Attribute | Description |
|---|---|
| `aaa-profile` | Defines global AAA profiles used for subscriber authentication, authorization, and accounting. |
| `access-profile` | Specifies global access profiles that control subscriber session behavior and policies. |
| `chassis-id` | Configures a unique chassis identifier for the node, used in subscriber session and redundancy handling. Range: 0 - 15. |
| `dhcp-relay` | Configures global DHCP relay settings to forward DHCP requests between clients and servers. |
| `dhcp-server` | Defines global DHCPv4 server configurations for dynamic IP address assignment to subscribers. |
| `dhcpv6-server` | Defines global DHCPv6 server configurations for IPv6 address and prefix assignment. |
| `interface` | Configures global access interface settings for handling subscriber traffic. |
| `l2tp-pool` | Defines global L2TPv2 tunnel endpoint pools used for subscriber session termination. |

| Attribute | Description |
|---|---|
| `l2tp-profile` | Specifies global L2TPv2 profiles that control tunnel parameters and behavior. |
| `pool` | Configures global IP address pools used for assigning addresses to subscribers. |
| `radius-profile` | Defines global RADIUS profiles for AAA communication parameters and behavior. |
| `radius-server` | Configures RADIUS server details used for subscriber authentication and accounting. |
| `service-profile` | Defines service profiles that apply policies and services to subscriber sessions. |
| `terminate-history-timeout` | Specifies the duration to retain terminated subscriber session history. Range: 0 - 720 hours. |
| `user-profile` | Defines user-specific profiles that control subscriber attributes and service parameters. |

The following options can be configured directly at the global access level, rather than within subsections.

| Attribute | Description |
|---|---|
| `chassis-id` | This option must be set to a unique value for each redundancy cluster and is only needed for stateful redundancy. Default: 0. |
| `terminate-history-timeout` | Change the retention period (in hours) for storing the termination history of subscribers after they have been disconnected. Default: 24. |

The following sections provides you the steps for each configuration. The process begins with setting up the access interface configuration, which is the first step. This is then followed by other important access profile and AAA profile configurations, which are crucial for managing subscriber access.

- Access Interface Configuration
- Access Profile Configuration
- AAA Profile Configuration

The second part covers the optional configurations:

- RADIUS Profile Configuration
- RADIUS Server Configuration
- Service Profile Configuration
- L2TP Profile Configuration
- Address Pool Configuration

The `user-profile` and `l2tp-pool` are the only components not referenced by name. The key here is

the user or pool name.

- User Profile Configuration
- L2TP Tunnel Pool Configuration

### Access Interface Configuration

While there is no single specific way to configure subscriber management, it is ideal to start with mandatory configurations and then move on to optional ones. The access interface configuration is the anchor point for almost all further access configurations. The interface configuration defines the access type and access profile. For more information, see sections Access Profile Configuration, `AAA profile` AAA Profile Configuration.

> Multiple interface configurations per IFP with disjoint VLAN ranges are supported.

The following image illustrates the access interface configurations and how they are associated with the entire subscriber management.

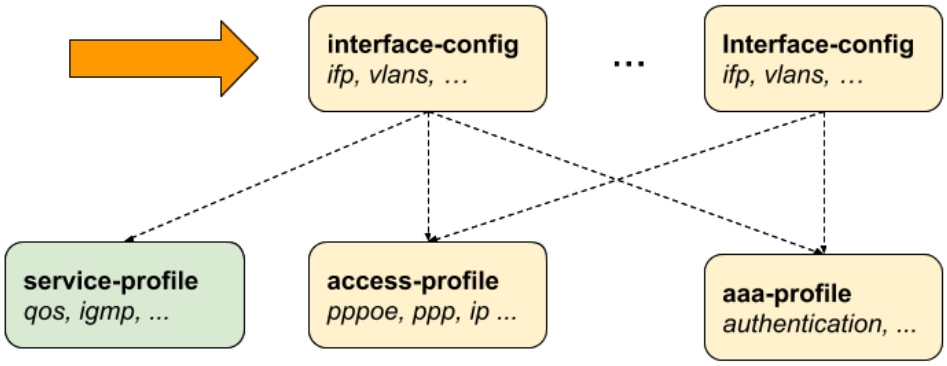



*Figure 66. Access Interface Configuration*

You can configure multiple interfaces for access and subscriber management, and each interface can reference the same profiles. This allows for efficient and scalable network management.

To configure the access interface, you must complete the following tasks.

1. Configure the physical interface name (IFP or LAG) and VLAN range
2. Configure the mandatory access type (PPPoE or IPoE)
3. Configure the mandatory access profile
4. Configure the mandatory AAA profile
5. Configure optional attributes such as service profile or session limit

#### Configuring Access Interfaces

This section describes how to configure access interfaces for subscriber management on the device. Access interfaces define how subscriber traffic is received, classified, and processed based on VLAN tagging and access type. Proper configuration ensures correct subscriber session establishment and policy application. It also enables integration with AAA, DHCP, and service profiles. Access interfaces can be configured without VLAN tags (untagged VLAN) and with one VLAN tag (single-

tagged) or with two VLAN tags (double-tagged).

**Syntax:**

```
set access interface <option>
```

The following table provides the command options/attributes and descriptions.

| Attribute | Description |
|---|---|
| `-` | Without any option, it configures an access interface. |
| `access-type` | Defines the access protocol used for this interface. This is a mandatory attribute. Values: PPPoE or IPoE. |
| `access-profile-name` | Specifies the name of the access profile (mandatory). For more information, see Access Profile Configuration. |
| `aaa-profile-name` | Specifies the name of the AAA profile (mandatory). For more information, see AAA Profile Configuration. |
| `service-profile-name` | This option allows assigning an optional service profile which can be dynamically overwritten via RADIUS. For more information, see Service Profile Configuration . |
| `max-subscribers-per-vlan` | This option defines the maximum number of subscribers per IFP and VLAN. A value of '1' will implicitly set the VLAN mode to 1:1 VLAN mode, where any value greater than 1 indicates N:1 VLAN mode. Default: 1 Range: 1 - 65535. |
| `max-subscribers-per-mac` | Maximum number of subscribers per IFP, VLAN, and MAC address. This option must be less or equal specified for the 'max-subscribers-per-vlan' attribute. This option does not affect IPoE subscribers, as they are inherently limited to a maximum of one per MAC address. Default: 1 Range: 1 - 65535. |
| `outer-vlan-encapsulation` | Limits the outer VLAN encapsulation to 802.1ad (TPID 0x88a8) or 802.1q (TPID 0x8100). By default, RBFS uses an auto-sensing mechanism that determines the correct TPID based on the first subscriber-initiated packet, such as PPPoE Discovery (PADI, PADR), DHCP discovery, or DHCPv6 solicit. Default: Auto-sensing. Values: 802.1ad or 802.1q. |
| `vlan-profile-enable` | If enabled, incoming PPPoE sessions (PPPoE PADI/PADR) are not honored unless matching vlan-profile is found in the table `global.vlan.profile` of the PPPoE daemon. For more information about VLAN profiles, see PPPoE VLAN Profiles. Default: False. |
| `gateway-ifl` | This options selects the IPoE gateway IFL (unnumbered source IFL) which is typically a loopback interface used as a gateway for IPoE subscribers. |

| Attribute | Description |
|---|---|
| `pppoe-pado-delay` | Specifies the PPPoE PADO delay, in seconds. The PPPoE PADO delay is used for stateless redundancy. For more information, PADO Delay-based Redundancy. |
| `dhcp-min-elapsed-time` | Defines the minimum elapsed time, in seconds, for DHCPv4/DHCPv6 transactions. This setting ensures that a DHCP request must meet the specified minimum duration before it is processed. Default: Disabled. Range: 1 - 600. The DHCPv4/v6 minimum elapsed time is used for stateless high availability. For more information, Stateless Redundancy for IPoE Subscribers. |
| `redundancy-session-id` | Specifies the redundancy session ID required for stateful high availability. Default: Disabled. Range: 1 - 65535. |

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Configuring Untagged Interfaces

This section explains how to configure interfaces that handle untagged subscriber traffic. Untagged interfaces are typically used in deployments where VLAN tagging is not required or is handled upstream. The configuration allows you to associate access profiles, AAA profiles, and subscriber policies directly with the physical interface. It is suitable for simple access deployments with minimal VLAN complexity.

**Syntax**:

**set access interface untagged** <interface-name> **aaa-profile-name** <aaa-profile-name>
**set access interface untagged** <interface-name> **access-profile-name** <access-profile-name>
**set access interface untagged** <interface-name> **access-type [PPPoE | IPoE]**
**set access interface untagged** <interface-name> **dhcp-min-elapsed-time** [1 - 600]
**set access interface untagged** <interface-name> **gateway-ifl** <gateway-ifl>
**set access interface untagged** <interface-name> **max-subscribers-per-mac** [1 - 65535]
**set access interface untagged** <interface-name> **max-subscribers-per-vlan** [1 - 65535]
**set access interface untagged** <interface-name> **pppoe-pado-delay** [1 - 255]
**set access interface untagged** <interface-name> **redundancy-session-id** [1 - 65535]
**set access interface untagged** <interface-name> **service-profile-name** <service-profile-name>
**set access interface untagged** <interface-name> **vlan-profile-enable [false | true]**

The following tables describes the command options to configure the untagged access interface.

| Attribute | Description |
|---|---|
| `interface-name` | Specifies the physical interface on which untagged subscriber access is configured. |
| `aaa-profile-name` | Specifies the AAA profile used for authentication, authorization, and accounting of subscribers. |

| Attribute | Description |
|---|---|
| access-profile-name | Associates an access profile that defines subscriber session handling and policies. |
| access-type | Specifies the access method for subscribers, such as PPPoE or IPoE. |
| dhcp-min-elapsed-time | Specifies the minimum elapsed time before accepting DHCP packets. Range: 1 - 600 seconds. The DHCPv4/v6 minimum elapsed time is used for stateless high availability. For more information, see [ngaccess:access_config_interface:::stateless-ha]. |
| gateway-ifl | Specifies the logical interface used as the gateway for IPoE subscribers. |
| max-subscribers-per-mac | Limits the number of subscriber sessions allowed per MAC address. Range: 1 - 65535. |
| max-subscribers-per-vlan | Limits the number of subscriber sessions allowed per VLAN. Range: 1 - 65535. |
| pppoe-pado-delay | Configures the delay before sending PPPoE PADO responses. Range: 1 - 255 seconds. |
| redundancy-session-id | Assigns a session ID used for subscriber session redundancy. Range: 1 - 65535. |
| service-profile-name | Specifies the service profile applied to subscriber sessions. |
| vlan-profile-enable | Enables or disables the use of VLAN profiles for subscriber management. |

Example:

In the following example configuration, the untagged access interface 'ifp-0/0/0' is configured for PPPoE access type, with the Access Profile set to 'pppoe-dual', Service Profile set to 'service-profile1', and AAA Profile set to 'aaa-radius'. The option 'vlan-profile-enable' is enabled by setting it to 'true', and the parameters 'max-subscribers-per-vlan' and 'max-subscribers-per-mac' are both assigned the value of '1'.

```
supervisor@switch: cfg> show config access interface untagged ifp-0/0/0
{
  "rtbrick-config:untagged": {
    "interface-name": "ifp-0/0/0",
    "access-type": "PPPoE",
    "access-profile-name": "pppoe-dual",
    "service-profile-name": "service-profile1",
    "aaa-profile-name": "aaa-radius",
    "vlan-profile-enable": "true",
    "max-subscribers-per-vlan": 1,
    "max-subscribers-per-mac": 1
  }
}
```

- Untagged interfaces are not supported on UfiSpace S9500-22XST (QAX) platform.

- The access `interface-name` can be a physical IFP or link aggregation (LAG).

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Configuring Single VLAN Tagged Interfaces

This section covers the configuration of interfaces that use a single VLAN tag (802.1Q) for subscriber identification. It allows defining a range of VLANs to support multiple subscribers over the same physical interface. This model is commonly used in broadband deployments where each VLAN represents a subscriber or service. The configuration supports flexible policy application per VLAN.

**set access interface single-tagged** <interface-name> <1-4094> <1-4094> **aaa-profile-name** <aaa-profile-name>
**set access interface single-tagged** <interface-name> <1-4094> <1-4094> **access-profile-name** <access-profile-name>
**set access interface single-tagged** <interface-name> <1-4094> <1-4094> **access-type [PPPoE | IPoE]**
**set access interface single-tagged** <interface-name> <1-4094> <1-4094> **dhcp-min-elapsed-time** [1 - 600]
**set access interface single-tagged** <interface-name> <1-4094> <1-4094> **gateway-ifl** <gateway-ifl>
**set access interface single-tagged** <interface-name> <1-4094> <1-4094> **max-subscribers-per-mac** [1 - 65535]
**set access interface single-tagged** <interface-name> <1-4094> <1-4094> **max-subscribers-per-vlan** [1 - 65535]
**set access interface single-tagged** <interface-name> <1-4094> <1-4094> **outer-vlan-encapsulation [802.1q | 802.1ad]**
**set access interface single-tagged** <interface-name> <1-4094> <1-4094> **pppoe-pado-delay** [1 - 255]
**set access interface single-tagged** <interface-name> <1-4094> <1-4094> **redundancy-session-id** [1 - 65535]
**set access interface single-tagged** <interface-name> <1-4094> <1-4094> **service-profile-name** <service-profile-name>
**set access interface single-tagged** <interface-name> <1-4094> <1-4094> **vlan-profile-enable [false | true]**

| Attribute | Description |
|---|---|
| `interface-name` | Specifies the physical interface on which subscriber access is configured. |
| `outer-vlan-min` | Defines the starting VLAN ID for the outer VLAN range. Range: 1 - 4094. |
| `outer-vlan-max` | Defines the ending VLAN ID for the outer VLAN range. Range: 1 - 4094. |
| `aaa-profile-name` | Specifies the AAA profile used for authentication, authorization, and accounting of subscribers. |
| `access-profile-name` | Associates an access profile that defines subscriber session handling and policies. |

| Attribute | Description |
|---|---|
| access-type | Specifies the access method for subscribers, such as PPPoE or IPoE. |
| dhcp-min-elapsed-time | Specifies the minimum elapsed time before accepting DHCP packets. Range: 1 - 600 seconds. The DHCPv4/v6 minimum elapsed time is used for stateless high availability. For more information, see [ngaccess:access_config_interface:::stateless-ha]. |
| gateway-ifl | Specifies the logical interface used as the gateway for IPoE subscribers. |
| max-subscribers-per-mac | Limits the number of subscriber sessions allowed per MAC address. Range: 1 - 65535. |
| max-subscribers-per-vlan | Limits the number of subscriber sessions allowed per VLAN. Range: 1 - 65535. |
| outer-vlan-encapsulation | Specifies the encapsulation type for the outer VLAN (for example, 802.1Q or 802.1ad). |
| pppoe-pado-delay | Configures the delay before sending PPPoE PADO responses. Range: 1 - 255 seconds. |
| redundancy-session-id | Assigns a session ID used for subscriber session redundancy. Range: 1 - 65535. |
| service-profile-name | Specifies the service profile applied to subscriber sessions. |
| vlan-profile-enable | Enables or disables the use of VLAN profiles for subscriber management. |

You need to define the VLAN identifier range within the range from 128 to 4000 for VLAN tagged interface. The following command and options are used to configure a single VLAN-tagged interface.

In the following example, the single tagged access interface ifp-0/0/0 is configured with the Outer VLAN minimum value `128` and the outer VLAN maximum value `3000`. The Access Type is defined PPPoE, Access Profile `pppoe-dual`, and AAA Profile as `aaa-radius`.

```
supervisor@switch: cfg> set access interface single-tagged ifp-0/0/0 128 3000 access-
type PPPoE
supervisor@switch: cfg> set access interface single-tagged ifp-0/0/0 128 3000 access-
profile-name pppoe-dual
supervisor@switch: cfg> set access interface single-tagged ifp-0/0/0 128 3000 aaa-
profile-name aaa-radius
supervisor@switch: cfg> commit
supervisor@switch: cfg> show config access interface single-tagged ifp-0/0/0 128 3000
{
  "rtbrick-config:single-tagged": [
    {
      "interface-name": "ifp-0/0/0",
      "outer-vlan-min": 128,
      "outer-vlan-max": 3000,
      "access-type": "PPPoE",
      "access-profile-name": "pppoe-dual",
      "aaa-profile-name": "aaa-radius"
```

```
    }
  ]
}
```

The access `interface-name` can be a physical IFP or LAG.

To access the RESTCONF API that corresponds to this CLI, click here.

#### Configuring Double Tagged VLAN Interfaces

This section describes how to configure interfaces using double VLAN tagging (Q-in-Q or 802.1ad). Double tagging enables hierarchical subscriber identification using outer (service) and inner (customer) VLANs. It is widely used in large-scale broadband networks to support multi-tenant or wholesale models. The configuration provides greater scalability and flexibility in subscriber and service separation.

When configuring double-tagged VLAN interfaces, setting the minimum and maximum VLAN ID to the same value ensures that only a specific VLAN ID is matched. It indicates that the VLAN interface will specifically recognize and process traffic tagged with that exact VLAN ID.

**Syntax:**

**set access interface double-tagged** <interface-name> <1-4094> <1-4094> <1-4094> <1-4094> **aaa-profile-name** <aaa-profile-name>
**set access interface double-tagged** <interface-name> <1-4094> <1-4094> <1-4094> <1-4094> **access-profile-name** <access-profile-name>
**set access interface double-tagged** <interface-name> <1-4094> <1-4094> <1-4094> <1-4094> **access-type** [**PPPoE** | **IPoE**]
**set access interface double-tagged** <interface-name> <1-4094> <1-4094> <1-4094> <1-4094> **dhcp-min-elapsed-time** [1 - 600]
**set access interface double-tagged** <interface-name> <1-4094> <1-4094> <1-4094> <1-4094> **gateway-ifl** <gateway-ifl>
**set access interface double-tagged** <interface-name> <1-4094> <1-4094> <1-4094> <1-4094> **max-subscribers-per-mac** [1 - 65535]
**set access interface double-tagged** <interface-name> <1-4094> <1-4094> <1-4094> <1-4094> **max-subscribers-per-vlan** [1 - 65535]
**set access interface double-tagged** <interface-name> <1-4094> <1-4094> <1-4094> <1-4094> **outer-vlan-encapsulation** [**802.1q** | **802.1ad**]
**set access interface double-tagged** <interface-name> <1-4094> <1-4094> <1-4094> <1-4094> **pppoe-pado-delay** [1 - 255]
**set access interface double-tagged** <interface-name> <1-4094> <1-4094> <1-4094> <1-4094> **redundancy-session-id** [1 - 65535]
**set access interface double-tagged** <interface-name> <1-4094> <1-4094> <1-4094> <1-4094> **service-profile-name** <service-profile-name>
**set access interface double-tagged** <interface-name> <1-4094> <1-4094> <1-4094> <1-4094>

**vlan-profile-enable [false | true]**

The following commands and options are used to configure double-tagged VLAN interfaces.

| Attribute | Description |
|---|---|
| `interface-name` | Specifies the physical interface on which double-tagged subscriber access is configured. |
| `outer-vlan-min` | Defines the starting VLAN ID for the outer VLAN range. Range: 1 - 4094. |
| `outer-vlan-max` | Defines the ending VLAN ID for the outer VLAN range. Range: 1 - 4094. |
| `inner-vlan-min` | Defines the starting VLAN ID for the inner VLAN range. Range: 1 - 4094. |
| `inner-vlan-max` | Defines the ending VLAN ID for the inner VLAN range. Range: 1 - 4094. |
| `aaa-profile-name` | Specifies the AAA profile used for authentication, authorization, and accounting of subscribers. |
| `access-profile-name` | Associates an access profile that defines subscriber session handling and policies. |
| `access-type` | Specifies the access method for subscribers, such as PPPoE or IPoE. |
| `dhcp-min-elapsed-time` | Specifies the minimum elapsed time before accepting DHCP packets. Range: 1 - 600 seconds. The DHCPv4/v6 minimum elapsed time is used for stateless high availability. For more information, see [ngaccess:access_config_interface:::stateless-ha]. |
| `gateway-ifl` | Specifies the logical interface used as the gateway for IPoE subscribers. |
| `max-subscribers-per-mac` | Limits the number of subscriber sessions allowed per MAC address. Range: 1 - 65535. |
| `max-subscribers-per-vlan` | Limits the number of subscriber sessions allowed per VLAN. Range: 1 - 65535. |
| `outer-vlan-encapsulation` | Specifies the encapsulation type for the outer VLAN (for example, 802.1Q or 802.1ad). |
| `pppoe-pado-delay` | Configures the delay before sending PPPoE PADO responses. Range: 1 - 255 seconds. |
| `redundancy-session-id` | Assigns a session ID used for subscriber session redundancy. Range: 1 - 65535. |
| `service-profile-name` | Specifies the service profile applied to subscriber sessions. |
| `vlan-profile-enable` | Enables or disables the use of VLAN profiles for subscriber management. |

In the following example, the double-tagged access interface `ifp-0/0/0` is configured with the Outer VLAN minimum value `128` and the outer VLAN maximum value `3000`. The configuration also defines the inner VLAN minimum value `7` and Inner VLAN maximum value `7`. The Access Type is defined PPPoE, Access Profile `pppoe-dual`, and AAA Profile `aaa-radius`.

```
supervisor@switch: cfg> set access interface double-tagged ifp-0/0/0 128 3000 7 7
```

```
access-type PPPoE
supervisor@switch: cfg> set access interface double-tagged ifp-0/0/0 128 3000 7 7
access-profile-name pppoe-dual
supervisor@switch: cfg> set access interface double-tagged ifp-0/0/0 128 3000 7 7 aaa-
profile-name aaa-radius
supervisor@switch: cfg> commit
supervisor@switch: cfg> show config access interface single-tagged ifp-0/0/0 128 3000
7 7
{
  "rtbrick-config:double-tagged": {
    "interface-name": "ifp-0/0/0",
    "outer-vlan-min": 128,
    "outer-vlan-max": 3000,
    "inner-vlan-min": 7,
    "inner-vlan-max": 7,
    "access-type": "PPPoE",
    "access-profile-name": "pppoe-dual",
    "aaa-profile-name": "aaa-radius"
  }
}
```

The access `interface-name` can be a physical IFP or LAG.

To access the RESTCONF API that corresponds to this CLI, click here.

### Access Profile Configuration

You must configure an access interface with an access profile name, such as `pppoe-dual` and it is also essential to configure the properties and parameters of the access profile itself.

The following image illustrates the access interface configurations and how they are associated with the entire subscriber management configurations.

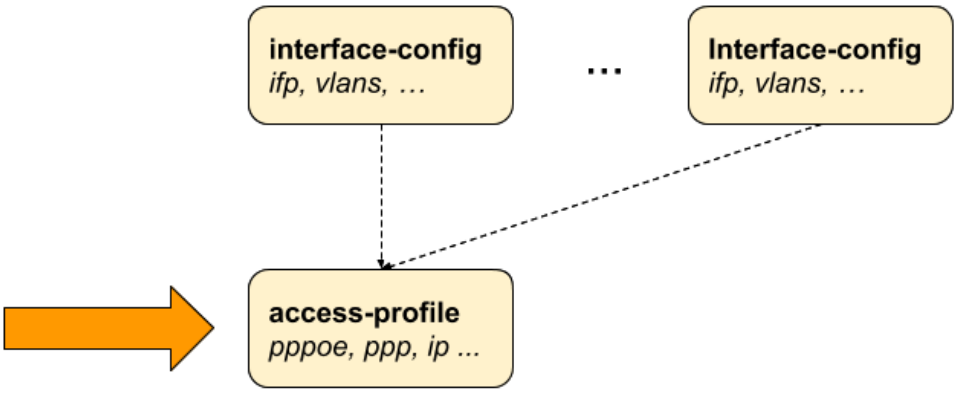



*Figure 67. Access Profile Configuration*

#### Configuring the Access Profile

Configuring an access profile involves specifying its name and defining various parameters to control how it handles network traffic and client interactions. The following command and options

allow you to configure an access profile.

**set access access-profile** <profile-name>
**set access access-profile** <profile-name> **address-family**
**set access access-profile** <profile-name> **instance**
**set access access-profile** <profile-name> **protocol**

The following table provides the command options/attributes and descriptions.

| Attribute | Description |
|---|---|
| `-` | Without any option, the command creates an access profile. |
| `<profile-name>` | Name of the access profile. |
| `instance` | Name of the outing instance. Default instance value is 'default'. The instance attribute in the access profile specifies the routing instance for PPPoE subscribers. For IPoE subscribers, the routing instance is determined by the 'gateway-ifl' parameter. |
| `address-family` | Address-family configuration |
| `protocol` | Protocol configuration. |

**Unrestricted Proxy-ARP Support for Subscriber IFLs Configuration**

This configuration option allows subscriber IFLs to enable unrestricted proxy-ARP, which responds to all ARP requests from subscribers using the MAC address of the corresponding subscriber IFL

**Syntax:**

**set access access-profile** <access-profile1> **address-family ipv4 proxy-arp** [**any** | **disabled** | **subscriber**]

| Attribute | Description |
|---|---|
| `<access-profile1>` | Specifies the name of the access profile. |
| `address-family ipv4` | Indicates that the configuration applies to the IPv4 address family. |
| `proxy-arp` | Enables and controls proxy ARP functionality for the access profile. |
| `any` | Allows the device to respond to ARP requests for any IP address, regardless of whether it belongs to a known subscriber. |
| `subscriber` | Allows the device to respond only to ARP requests for known subscriber IP addresses. |
| `disabled` | Disables Proxy ARP functionality for the access profile. |

Example:

This configuration creates an access profile that enables DHCP (server mode), DHCPv6 (server mode), and IPv6 Router Advertisements (RA) for subscriber address assignment. It uses the specified IPv4 and IPv6 address pools, supports IPv6 prefix delegation through a dedicated pool, advertises the configured IPv6 DNS server, and enables IPv4 proxy ARP with the 'any' option.

```
supervisor@rtbrick: cfg> show config access access-profile access-profile1
{
  "rtbrick-config:access-profile": [
    {
      "profile-name": "access-profile1",
      "protocol": {
        "ra": {
          "enable": "true"
        },
        "dhcp": {
          "enable": "true",
          "mode": "server"
        },
        "dhcpv6": {
          "enable": "true",
          "mode": "server"
        }
      },
      "address-family": {
        "ipv4": {
          "enable": "true",
          "pool-name": "pool1",
          "proxy-arp": "any"
        },
        "ipv6": {
          "enable": "true",
          "pool-name": "pool1",
          "prefix-delegation-pool-name": "pool2",
          "primary-dns": "1::1"
        }
      }
    }
  ]
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Configuring IPv4

To enable IPv4 support for access protocols like PPP (PPPoE) or DHCP (IPoE), explicitly activate the IPv4 address family in the access profile.

**Syntax:**

**set access access-profile** <profile-name> **address-family ipv4 dad-enable [false | true]**
**set access access-profile** <profile-name> **address-family ipv4 enable [false | true]**
**set access access-profile** <profile-name> **address-family ipv4 pool-name [pool - name]**
**set access access-profile** <profile-name> **address-family ipv4 primary-dns [primary - dns]**
**set access access-profile** <profile-name> **address-family ipv4 proxy-arp [disabled | subscriber | any]**
**set access access-profile** <profile-name> **address-family ipv4 proxy-arp-enable [false | true]**
**set access access-profile** <profile-name> **address-family ipv4 secondary-dns [secondary-dns]**
**set access access-profile** <profile-name> **address-family ipv4 static-ipv4 [static-ipv4]**

The following table provides the command options/attributes and descriptions.

| Attribute | Description |
|---|---|
| `enable` | Enable IPv4. Default: false. |
| `pool-name` | The `pool-name` option (optional) allows assigning the IPv4 address from a locally managed pool. For more details, see Address Pool Configuration. This address is used by protocols like PPP IPCP (PPPoE) or DHCP (IPoE) as a client or peer IPv4 address. |
| `primary-dns` | Primary DNS server. |
| `secondary-dns` | The primary DNS and secondary DNS servers configured are used by protocols such as PPP (PPPoE) or DHCP (IPoE) and advertised to the client. |
| `proxy-arp-enable` | Enable (or disable) proxy ARP support for IPoE subscribers. When proxy ARP is enabled, if the RBFS device receives an ARP request from the subscriber for which it has a route to the target (destination) IP address, the RBFS device responds by sending a proxy ARP reply packet containing its own MAC address. The host/subscriber that sent the ARP request then sends the actual destined packets to RBFS, which forwards them to the intended destination. Default: NONE. |
| `static-ipv4` | Configures the static IPv4 address to a subscriber. Multiple subscriber in the same instance cannot be configured with same static IPv4 address.<br>This feature should be used with caution. |

This configuration specifies the primary IPv4 DNS server that is advertised to subscribers using the access profile.

```
supervisor@rtbrick.net: cfg> show config access access-profile ipoe-intf2-st-default-
profile add
ress-family ipv4 primary-dns
```

```
{
  "rtbrick-config:primary-dns": "10.1.1.1"
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Configuring IPv6

To make IPv6 available for access protocols such as PPP (PPPoE) or DHCP (IPoE), you must explicitly enable the IPv6 address family within the access profile.

**Syntax**:

**set access access-profile** <profile-name> **address-family ipv6 dad-enable [false** | **true]**
**set access access-profile** <profile-name> **address-family ipv6 enable [false** | **true]**
**set access access-profile** <profile-name> **address-family ipv6 pool-name** <pool-name>
**set access access-profile** <profile-name> **address-family ipv6 prefix-delegation-pool-name** <prefix-delegation-pool-name>
**set access access-profile** <profile-name> **address-family ipv6 primary-dns** <primary-dns>
**set access access-profile** <profile-name> **address-family ipv6 secondary-dns** <secondary-dns>

The following table provides the command options/attributes and descriptions.

| Attribute | Description |
|---|---|
| `enable` | Enable IPv6. Default: False. |
| `pool-name` | Local IPv6 pool name. |
| `prefix-delegation-pool-name` | The `pool-name` attribute (optional) allows to assign of the IPv6 prefix from a locally managed pool. For more details, see Address Pool Configuration. This prefix is advertised by ICMPv6 router advertisements to the client where prefixes from optional `prefix-delegation-pool-name` are advertised by DHCPv6 as delegated prefix (IA_PD). |
| `primary-dns` | Primary DNS server. |
| `secondary-dns` | The primary DNS and secondary DNS servers configured are used by protocols such as ICMPv6 router advertisements or DHCPv6 and advertise to the client. |

This configuration assigns the specified IPv6 address pool (ipoe_v6_pool1) to the access profile for subscriber address allocation.

```
supervisor@rtbrick.net: cfg> show config access access-profile ipoe-intf2-st-default-
profile add
ress-family ipv6 pool-name
{
```

```
  "rtbrick-config:pool-name": "ipoe_v6_pool1"
}
```

This configuration enables the DHCP protocol for the access profile and sets the DHCP lease time to 30.

```
supervisor@rtbrick.net: cfg> show config access access-profile ipoe-intf2-st-default-
profile protocol dhcp
{
  "rtbrick-config:dhcp": {
    "enable": "true",
    "lease-time": 30
  }
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Enabling IPv6 Router Advertisement

IPv6 Router Advertisement (RA) is a key function within the IPv6 Neighbor Discovery Protocol (NDP). Routers use RA messages to announce their presence and share essential network parameters with IPv6-capable devices. Enabling Router Advertisement is critical for the automatic configuration of IPv6 addresses and related settings on devices supporting IPv6.

**set access access-profile** <profile-name> **protocol ra enable** [**false** | **true**]
**set access access-profile** <profile-name> **protocol ra interval** [0 - 255]
**set access access-profile** <profile-name> **protocol ra lifetime** [0 - 4294967295]
**set access access-profile** <profile-name> **protocol ra preferred-lifetime** [0 - 4294967295]
**set access access-profile** <profile-name> **protocol ra router-lifetime** [0 - 9000]

The following table provides the command options/attributes and descriptions.

| Attribute | Description |
|---|---|
| `enable` | Enable IPv6 router advertisement. Default: False. |
| `interval` | IPv6 router advertisements interval in seconds. Sets the interval at which the Router Advertisement messages are sent. Default: 0 ('0' indicates disabled state). |
| `lifetime` | The valid lifetime for the prefix in seconds. Default: 14400. |
| `preferred-lifetime` | The preferred lifetime for the prefix in seconds. Default: 1800. |

#### Configuring DHCPv4

To enable and configure DHCPv4 in the access profile, use the following command. This configuration enables the access profile to process DHCP requests and assign IP addresses to clients.

**Syntax**:

```
set access access-profile <profile-name> protocol dhcp dhcp-server <dhcp-server>
set access access-profile <profile-name> protocol dhcp enable [false | true]
set access access-profile <profile-name> protocol dhcp lease-time [1 - 4294967295]
set access access-profile <profile-name> protocol dhcp mode [server | relay]
```

The following table provides the command options and descriptions.

| Attribute | Description |
|---|---|
| `<cr>` | With any option, the command sets the DHCP mode for the specified access profile. |
| `enable` | True or false. Specify true to enable DHCP. Default: False. |
| `dhcp-mode` | This option specifies the DHCP mode for handling DHCP requests. Default: server. Values: server and relay. |
| `lease-time` | Define the duration of IP address leases, in seconds. Default: 300 |
| `dhcp-server` | Configure global DHCP server. |

API To access the RESTCONF API that corresponds to this CLI, click here.

To ensure consistency, it is highly recommended to align the DHCPv4 lease duration with the DHCPv6 lifetime, while also synchronizing the corresponding renewal and rebind intervals.

#### Configuring DHCPv6

To configure DHCPv6 in an access profile, use the following command. This configuration activates DHCPv6 functionality, sets prefix lifetimes, and defines the operating mode.

```
set access access-profile <profile-name> protocol dhcpv6 dhcpv6-server <dhcpv6-server>
set access access-profile <profile-name> protocol dhcpv6 enable [false | true]
set access access-profile <profile-name> protocol dhcpv6 lifetime [30 - 4294967295]
set access access-profile <profile-name> protocol dhcpv6 mode [server | relay]
set access access-profile <profile-name> protocol dhcpv6 preferred-lifetime [30 -
4294967295]
set access access-profile <profile-name> protocol dhcpv6 rapid-commit [false | true]
set access access-profile <profile-name> protocol dhcpv6 wait-for-dhcpv4 [false | true]
```

The following table provides the command options/attributes and descriptions.

| Attribute | Description |
|---|---|
| `enable` | Enable DHCPv6. Default: False. |

| Attribute | Description |
|---|---|
| `mode` | This option defines the DHCPv6 mode where the server handles DHCPv6 requests locally and relay/proxy forwards those to the configured servers. The difference between relay and proxy is that proxy can hide the actual DHCPv6 server. Default: server. Values: server and relay. |
| `lifetime` | The duration for which the advertised prefix is valid. This is the total time the prefix remains usable before it needs to be renewed. Set this to the desired lifetime in seconds. Default: 14400. |
| `preferred-lifetime` | Set the lifetime for IPv6 prefixes, in seconds. It defines the duration during which the prefix is preferred for use. It should be less than or equal to the lifetime. If set higher than the lifetime, it will be adjusted to match the lifetime.<br><br>The values for T1 and T2 are 0.5 and 0.8 times the shortest preferred lifetime. Default: 1800. |
| `dhcpv6-server` | Configure DHCPv6 server. |
| `rapid-commit [false \| true]` | Specify `true` to enable DHCPv6 rapid commit. Specify `false` to disable support for the DHCPv6 rapid commit option for the access profile. Default value: False. For more information, see DHCPv6 Rapid Commit Control. |
| `wait-for-dhcpv4 [false \| true]` | Specify `true` to wait for DHCPv4 to complete before processing DHCPv6. Specify `false` to process DHCPv6 independently of DHCPv4. Default value: False. For more information, see Dual-Stack Splitting Prevention in IPoE Stateless Redundancy Deployments. |

To access the RESTCONF API that corresponds to this CLI, click here.

#### Configuring PPPoE and PPP

The PPPoE protocol must be explicitly enabled in the access profile to allow PPPoE sessions.

#### PPPoE

The PPPoE configuration allows changing the default behavior of the PPPoE protocol.

**set access access-profile** <profile-name> **protocol pppoe enable [false | true]**
**set access access-profile** <profile-name> **protocol pppoe max-outstanding** <1 - 65535>
**set access access-profile** <profile-name> **protocol pppoe session-protection enable [false | true]**
**set access access-profile** <profile-name> **protocol pppoe session-protection max-lockout** [1 - 65535]
**set access access-profile** <profile-name> **protocol pppoe session-protection min-lockout** [1 - 255]
**set access access-profile** <profile-name> **protocol pppoe session-protection min-uptime** <1 - 65535>

Sample command:

```
set access access-profile pppoe-dual protocol pppoe enable true
```

The following table provides the command options/attributes and descriptions.

| Attribute | Description |
|---|---|
| `-` | Without any option, the command sets the access protocol. |
| `enable` | Enable PPPoE. Default: False. |
| `max-outstanding` | Maximum outstanding PPPoE sessions. Default: 512. Range: 1 - 65535. |
| `session-protection` | PPPoE session protection.<br><br>When PPPoE session protection is enabled, any short-lived or failed sessions are logged. By default, a session that does not remain established for at least 60 seconds (min-uptime) is classified as a failed or short-lived session. Such failures trigger a block on new sessions for this IFP, VLAN, and MAC for one second (min-lockout) by default. With each subsequent failed session, the lockout duration increases exponentially until it reaches a maximum of 300 seconds (max-lockout). If no failed sessions occur for 900 seconds, the lockout interval is reset (currently, not configurable).<br><br>PPPoE session protection also logs the last subscriber ID and terminates the session with a code that indicates the reason for the failure. |

| Attribute | Description |
|---|---|
| `<cr>` | Without any option, the command sets session protection for the access profile. |
| `enable` | Enables PPPoE session protection. Default: false. |
| `min-lockout` | Session protection minimum lockout time (in seconds). Default: 1. |
| `max-lockout` | Session protection maximum lockout time (in seconds). Default: 300. |
| `min-uptime` | Session with an uptime less than this will trigger protection (in seconds). Default: 60. |

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Configuring PPP LCP

The PPP Link Control Protocol (LCP) configuration allows changing the default behavior of the LCP protocol.

**Syntax**:

**set access access-profile** <profile-name> **protocol ppp lcp authentication-protocol [PAP** |

```
CHAP | PAP_CHAP | CHAP_PAP]
set access access-profile <profile-name> protocol ppp lcp config-nak-max [1 - 255]
set access access-profile <profile-name> protocol ppp lcp echo-enable [false | true]
set access access-profile <profile-name> protocol ppp lcp echo-interval [1 - 255]
set access access-profile <profile-name> protocol ppp lcp echo-max-retransmit [1 - 255]
set access access-profile <profile-name> protocol ppp lcp lcp-loop-detection [false | true]
set access access-profile <profile-name> protocol ppp lcp mru [1280 - 1500]
set access access-profile <profile-name> protocol ppp lcp mru-negotiation [false | true]
set access access-profile <profile-name> protocol ppp lcp mtu [1280 - 1500]
set access access-profile <profile-name> protocol ppp lcp retransmit-interval [1 - 255]
set access access-profile <profile-name> protocol ppp lcp retransmit-max [1 - 255]
```

The following table provides the command options and description.

| Attribute | Description |
|---|---|
| `authentication-protocol` | This option allows you to specify the authentication protocol used during the LCP negotiation phase. By default, PPP authentication is set to `PAP_CHAP`. This can be changed by setting the authentication protocol to either `PAP`, `CHAP`, `PAP_CHAP` or `CHAP_PAP`. The Password Authentication Protocol (PAP) is (defined in RFC 1334) and receives the password as a plain text value from the client. The Challenge Handshake Authentication Protocol (CHAP) is (defined in RFC 1994) provides a more secure way to authenticate the client without exchanging plaintext secrets. The `PAP_CHAP` option first attempts to authenticate using `PAP` with a fallback to `CHAP`, if `PAP` is rejected by the client. Alternatively, CHAP_PAP starts with CHAP and falls back to PAP if CHAP is rejected by the client. Default: PAP_CHAP. |
| `echo-enable` | By default, RBFS responds to LCP echo requests but does not initiate them unless `echo-enable` is set to true. Default: true |
| `echo-interval` | LCP echo request interval in seconds. Default: 30, Range: 1 - 255. |
| `echo-max-retransmit` | LCP echo request retransmissions. Default: 3, Range: 1 - 255. |
| `mru-negotiation` | Negotiate MRU. Default: true. |
| `mru` | Maximum local MRU (peer MTU). Default: 1492, Range: 1280 - 1500. |
| `mtu` | Maximum local MTU (peer MRU). Default: 1492, Range: 1280 - 1500. |
| `lcp-loop-detection` | This is used during the PPP connection process to detect and prevent loops in the network. It is achieved through the negotiation and validation of 'magic numbers'. Magic numbers are unique values used by both ends of a PPP connection to ensure that the data sent is not simply being looped back from the other end. By default, the negotiation and validation of magic numbers are enabled, providing protection against looping connections. You can disable loop detection by setting `lcp-loop-detection` to false. However, it is NOT recommended to disable LCP loop detection. Default: true. |
| `retransmit-interval` | The LCP request retransmission interval. Default: 5, Range: 1 - 255. |

| Attribute | Description |
|---|---|
| `retransmit-max` | The LCP requests retransmission before the session is terminated if no response is received. Default: 3, Range: 1 - 255. |
| `config-nak-max` | The option `config-nak-max` defines the maximum PPP LCP configuration `reject/nak` messages that can be sent or received before the session is terminated. Default: 16, Range: 1 - 255. |

To access the RESTCONF API that corresponds to this CLI, click here.

#### Configuring PPP IPCP

To configure IPv4 over PPPoE, explicitly enable both the address-family IPv4 and the PPP IPCP protocol in the access profile. The `source-ifl` is a mandatory option that specifies the logical interface from which the local IPv4 address will be derived.

**Syntax**:

**set access access-profile** <profile-name> **protocol ppp ipcp config-nak-max** [1 - 255]
**set access access-profile** <profile-name> **protocol ppp ipcp enable [false** | **true]**
**set access access-profile** <profile-name> **protocol ppp ipcp passive [false** | **true]**
**set access access-profile** <profile-name> **protocol ppp ipcp retransmit-interval** [1 - 255]
**set access access-profile** <profile-name> **protocol ppp ipcp retransmit-max** [1 - 255]
**set access access-profile** <profile-name> **protocol ppp ipcp source-ifl** <source-ifl>

The following table provides the command options/attributes and descriptions.

| Attribute | Description |
|---|---|
| `enable` | Enable IPCP. Default: False. |
| `passive` | IPCP passive mode. Default: False. |
| `source-ifl` | This setting is required and should be set to obtain the local IPv4 address from a specific logical interface. It should be configured to use the loopback interface of the corresponding routing instance. If a `source-ifl` is chosen from a different routing instance, it will not change the routing instance assigned to the subscriber. Also, the routing instance can be changed through a RADIUS `Access-Accept` message without affecting the `source-ifl`. This means that the `source-ifl` can still be used to obtain the local IPv4 address, even if it belongs to a different routing instance. Although, it is not mandatory, it is recommended to configure the same loopback address across all routing instances where a subscriber might be located. |
| `retransmit-interval` | The IPCP request retransmission interval. Default: 5, Range: 1 - 255. |
| `retransmit-max` | The IPCP requests retransmission before the session is terminated if no response is received. Default: 8 Range: 1 - 255. |

| Attribute | Description |
|---|---|
| `config-nak-max` | The option config-nak-max defines the maximum PPP IPCP configuration reject/nak messages that can be sent or received before the session is terminated. Default: 8, Range: 1 - 255. |

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Configuring PPP IP6CP

To enable IPv6 over PPPoE, you must explicitly configure both the IPv6 address family and the PPP IPv6 Control Protocol (IP6CP).

**set access access-profile** <profile-name> **protocol ppp ip6cp config-nak-max** [1 - 255]
**set access access-profile** <profile-name> **protocol ppp ip6cp enable** [**false** | **true**]
**set access access-profile** <profile-name> **protocol ppp ip6cp passive** [**false** | **true**]
**set access access-profile** <profile-name> **protocol ppp ip6cp retransmit-interval** [1 - 255]
**set access access-profile** <profile-name> **protocol ppp ip6cp retransmit-max** [1 - 255]

The following command demonstrates how to enable PPP IP6CP in the `pppoe-dual` access profile.

The following table provides the command options/attributes and descriptions.

| Attribute | Description |
|---|---|
| `enable` | Enable IP6CP. Default: false. |
| `passive` | IP6CP passive mode. Default: false. |
| `retransmit-interval` | This option sets the interval, in seconds, between retransmissions, for IP6CP requests. Default: 5 Range: 1 - 255. |
| `retransmit-max` | This option sets the maximum number of retransmission attempts. The IP6CP requests retransmission before the session is terminated if no response is received. Default: 8, Range: 1 - 255. |
| `config-nak-max` | The option defines the maximum PPP IP6CP configuration `reject/nak` (Negative Acknowledgement) messages that can be sent or received before the session is terminated. Default: 6, Range: 1 - 255. |

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Configuring IPoE Username and Password Options for Access Profiles

You can configure the IPoE username suffix and password in both encrypted and plain-text formats.

**Syntax:**

**set access access-profile** <profile-name> **protocol ipoe password-encrypted-text** <password-encrypted-text[4-16419]>

**set access access-profile** <profile-name> **protocol ipoe password-plain-text** <password-plain-text[1-64]>
**set access access-profile** <profile-name> **protocol ipoe username-suffix** <username-suffix[0-64]>

The following table provides the command options/attributes and descriptions.

| Option | Description |
|---|---|
| `<password-encrypted-text>` | Specifies the encrypted password for IPoE subscribers. The value must be between 4 and 16419 characters in length. |
| `<password-plain-text>` | Specifies the plain text password for IPoE subscribers. The value must be between 1 and 64 characters in length. |
| `<username-suffix>` | Specifies the suffix for the IPoE username. By specifying a suffix, the username suffix can be modified from its standard `@ipoe` value. Setting it to an empty string ("") enables the use of the MAC address alone. The value must be between 0 and 64 characters in length. For example, if you configure `test`, the resulting username will be changed to <MAC>#test#. |

The following example shows how to set the IPoE password in encrypted text format and the username suffix for an access profile.

```
supervisor@rtbrick.net: cfg> show conf set access access-profile
set access access-profile profile1
set access access-profile profile1 protocol ipoe username-suffix #test
set access access-profile profile1 protocol ipoe password-encrypted-text
$1a00367e016f343e121d3eb7d573c02ae
```

```
supervisor@rtbrick.net: cfg> show conf access access-profile
{
  "rtbrick-config:access-profile": [
    {
      "profile-name": "profile1",
      "protocol": {
        "ipoe": {
          "username-suffix": "#test",
          "password-encrypted-text": "$1a00367e016f343e121d3eb7d573c02ae"
        }
      }
    }
  ]
}
```

#### Retention Timer for Subscriber Termination History Configuration

You can configure a retention period (in hours) for storing the termination history of access

subscribers after they have been disconnected.

**Syntax**:

**set access terminate-history-timeout** [0 - 720]

| Option | Description |
|---|---|
| `<value>` | Specify the period of time (in hours). Range: 0 - 720. If '0' is specified, the data is not retained. |

This configuration sets the terminate-history-timeout to 10, defining the duration the system retains information about terminated subscriber sessions.

```
set access terminate-history-timeout 10
```

```
{
  "rtbrick-config:access": {
    "terminate-history-timeout": 10
  }
}
```

#### Configuring DHCP Asymmetric Lease

You can configure the DHCP Asymmetric Lease feature for both DHCPv4 and DHCPv6 in relay mode.

**Syntax**:

**set access access-profile** <profile-name> **protocol dhcp asymmetric** (**false**| **true**)
**set access access-profile** <profile-name> **protocol dhcp rebind-time** [1 - 4294967295]
**set access access-profile** <profile-name> **protocol dhcp renewal-time** [1 - 4294967295]
**set access access-profile** <profile-name> **protocol dhcpv6 asymmetric** (**false** | **true**)
**set access access-profile** <profile-name> **protocol dhcpv6 rebind-time** [0 - 4294967295]
**set access access-profile** <profile-name> **protocol dhcpv6 renewal-time** [0 - 4294967295]

The following table provide the command options and descriptions.

| Attribute | Description |
|---|---|
| `<profile-name>` | Name of the access profile. |
| `asymmetric [false \| true]` | Specify 'true' to enable the asymmetric lease feature. By default, this is false. |

| Attribute | Description |
|---|---|
| `rebind-time [1 - 4294967295]` | Specifies the DHCP rebind time that is sent to DHCP clients. If lease renewal is unsuccessful, the client waits until the timer expires and then attempts to rebind the lease. Range: 1 - 4294967295. |
| `renewal-time [1-4294967295]` | Defines when a client should attempt to renew its existing lease time with the original DHCP server. Range: 1 - 4294967295. |
| `dhcpv6 rebind-time [0 - 4294967295]` | Specifies the DHCPv6 rebind timer that will be advertised to the client. Range: 0 - 4294967295. |
| `dhcpv6 renewal-time [0 - 4294967295]` | Specifies the DHCPv6 renewal timer that will be advertised to the client. Range: 0 - 4294967295. |

**Example Scenario:**

Server Assignment: A DHCPv6 server assigns a lifetime of 360 seconds, a renewal timer of 240 seconds, and a rebind timer of 300 seconds.

RBFS Configuration: RBFS is configured with Asymmetric Lease enabled and overwrites only the renewal time to 30 seconds.

Result: Because the client attempts to renew every 30 seconds, RBFS will locally serve the first 5 renewal requests (up to 150 seconds). The 6th renewal occurs at 180 seconds, exactly 50% of the 360-second lifetime, prompting RBFS to forward this request to the upstream server.

The following are example configurations for DHCPv4 and DHCPv6.

This configuration enables DHCPv4 relay for subscribers using the default-profile access profile and forwards DHCPv4 messages to the server 'dhcp_server2'. The asymmetric value is set to 'true'. Assigned IPv4 addresses have a lease time of 360 seconds. The CPE tries to renew its lease after 60 seconds and, if unsuccessful, enters the rebind phase after 300 seconds.

```
set access access-profile default-profile protocol dhcp enable true
set access access-profile default-profile protocol dhcp mode relay
set access access-profile default-profile protocol dhcp asymmetric true
set access access-profile default-profile protocol dhcp lease-time 360
set access access-profile default-profile protocol dhcp renewal-time 60
set access access-profile default-profile protocol dhcp rebind-time 300
set access access-profile default-profile protocol dhcp dhcp-server dhcp_server2
```

```
{
  "rtbrick-config:dhcp": {
    "enable": "true",
    "mode": "relay",
    "asymmetric": "true",
    "lease-time": 360,
    "renewal-time": 60,
    "rebind-time": 300,
```

```
    "dhcp-server": [
      "dhcp_server2"
      ]
  }
}
```

DHCPv6:

This configuration enables DHCPv6 relay for subscribers using the 'default-profile' access profile and forwards DHCPv6 messages to the server 'dhcp_server2'. The asymmetric value is set to 'true'. Assigned IPv6 addresses have a valid lifetime of 360 seconds and a preferred lifetime of 320 seconds. The CPE tries to renew its lease after 60 seconds and, if unsuccessful, enters the rebind phase after 300 seconds.

```
set access access-profile default-profile protocol dhcpv6 enable true
set access access-profile default-profile protocol dhcpv6 mode relay
set access access-profile default-profile protocol dhcpv6 asymmetric true
set access access-profile default-profile protocol dhcpv6 lifetime 360
set access access-profile default-profile protocol dhcpv6 preferred-lifetime 320
set access access-profile default-profile protocol dhcpv6 renewal-time 60
set access access-profile default-profile protocol dhcpv6 rebind-time 300
set access access-profile default-profile protocol dhcpv6 dhcpv6-server dhcp_server2
```

```
{
  "rtbrick-config:dhcpv6": {
    "enable": "true",
    "mode": "relay",
    "asymmetric": "true",
    "lifetime": 360,
    "preferred-lifetime": 320,
    "renewal-time": 60,
    "rebind-time": 300,
    "dhcpv6-server": [
      "dhcp_server2"
      ]
  }
}
```

**DHCPv6 Rapid Commit Control**

By default, DHCPv6 permits the Rapid Commit option (number 14) included in client requests and completes address assignment using the two-message Rapid Commit exchange. DHCPv6 Rapid Commit Control feature allows the DHCPv6 server to ignore the Rapid Commit option included in client requests. When enabled, the server forces clients to use the standard four-message DHCPv6 exchange instead of the two-message Rapid Commit process. You can enable or disable DHCPv6 Rapid Commit for subscribers using a specific access profile.

Example:

The following configuration enables the DHCPv6 server, enables DHCPv6 Rapid Commit, and sets the DHCPv6 lease lifetime to 90 seconds.

```
set access access-profile ipoe protocol dhcpv6 enable true
set access access-profile ipoe protocol dhcpv6 rapid-commit true
```

```
{
  "rtbrick-config:dhcpv6": {
    "enable": "true",
    "rapid-commit": "true",
    "lifetime": 90
  }
}
```

**Example for PPPoE with IPv4 and IPv6**

The following access profile configuration is a PPPoE access profile named `pppoe-dual` that supports both IPv4 and IPv6. The instance is marked as `default`, and the session protection is enabled for PPPoE by setting the value to `true`, and the VLAN priority is set to `6`.

The configuration defines various PPP parameters such as LCP, IPCP, and so on. For LCP (Link Control Protocol), the authentication protocol is specified as `PAP_CHAP`, indicating that both `PAP` (Password Authentication Protocol) and `CHAP` (Challenge Handshake Authentication Protocol) are allowed. The `echo-interval`, which is the interval in seconds for sending LCP echo requests to check the connection, is specified as 30 seconds. The `echo-max-retransmit` parameter, which is the maximum number of retransmissions for LCP echo requests before considering the link down, is set to `3`. The configuration enables the LCP echo requests by specifying true for the `echo-enable` parameter.

In the configuration, both the IPCP and IP6CP are enabled by setting their values to `true`. These protocols are used to configure both the IPv4 and IPv6 settings over PPP. The source IFL (logical interface) for IPCP is specified as 'lo-0/0/0/1'. The RA (Router Advertisement) Configuration is enabled by setting the value to 'true', which allows the router to send RA messages for IPv6 configuration. The `interval` period for the RA messages is set to 60 seconds. DHCPv6 is enabled by setting the value to `true`, which allows the device to assign IPv6 addresses to the subscribers through DHCP.

The L2TP tunnel profile is defined as `l2tp-default`, which allows the encapsulation of PPP sessions over IP networks. Both IPv4 and IPv6 address families are enabled by setting their values to `true` using the address family configurations. For IPv4, the primary DNS server is specified '198.51.100.1' and the secondary DNS server is specified '198.51.100.4'. For IPv6, the primary DNS server is specified as '2001:db8:0:100::'' and the secondary DNS server is specified '2001:db8:0:104::'.

```
supervisor@switch: cfg> show config access access-profile pppoe-dual
{
  "rtbrick-config:access-profile": {
    "profile-name": "pppoe-dual",
```

```
    "instance": "default",
    "protocol": {
      "pppoe": {
        "enable": "true",
        "session-protection": {
          "enable": "true"
        },
        "vlan-priority": 6
      },
      "ppp": {
        "lcp": {
          "authentication-protocol": "PAP_CHAP",
          "echo-interval": 30,
          "echo-max-retransmit": 3,
          "echo-enable": "true"
        },
        "ipcp": {
          "enable": "true",
          "source-ifl": "lo-0/0/0/1"
        },
        "ip6cp": {
          "enable": "true"
        }
      },
      "ra": {
        "enable": "true",
        "interval": 60
      },
      "dhcpv6": {
        "enable": "true"
      },
      "l2tp": {
        "tunnel-profile": "l2tp-default"
      }
    },
    "address-family": {
      "ipv4": {
        "enable": "true",
        "primary-dns": "198.51.100.1",
        "secondary-dns": "198.51.100.4"
      },
      "ipv6": {
        "enable": "true",
        "primary-dns": "2001:db8:0:100::",
        "secondary-dns": "2001:db8:0:104::"
      }
    }
  }
}
```

**Example for IPoE with IPv4 and IPv6**

The example configuration below sets up an IPoE access profile named 'ipoe-dual' to handle IPoE sessions, which support both IPv4 and IPv6 address families. The router acts as a DHCP server for both protocols, assigning IP addresses and DNS server information to client devices.

The profile name is `ipoe-dual`. By setting the value to `true`, this profile enables both DHCP and DHCPv6 protocols. The mode is defined as `server` for both DHCP and DHCPv6, which allows the router to act as a server for IPv4 and IPv6. The IPv4 address family is configured for the access profile by setting the value to `true`. For IPv4, proxy ARP is enabled by setting `proxy-arp-enable` to true.

The IPv4 address family is configured for the access profile by setting the value to ‘`true`. For IPv4, proxy ARP is enabled by setting `proxy-arp-enable` to true. This allows the router to respond to ARP requests on behalf of other hosts.

The pool name `ipoe` indicates the IPv4 address allocation to client devices. The primary DNS server for IPv4 is set to '198.51.100.1', and the secondary DNS server is specified as '198.51.100.4'.

The configuration includes various parameters for IPv6. The IPv6 pool name is `ipoe-ia-na`, which provides IPv6 addresses to subscribers. The `prefix-delegation-pool-name` parameter is set as `ipoe-ia-pd` for IPv6 prefix delegation. The primary and secondary DNS servers for IPv6 are specified as '2001:db8:0:100::' and '2001:db8:0:104::' respectively.

```
supervisor@switch: cfg> show config access access-profile ipoe-dual
{
  "rtbrick-config:access-profile":{
    "profile-name":"ipoe",
    "protocol":{
      "dhcp":{
        "enable":"true",
        "mode":"server"
      },
      "dhcpv6":{
        "enable":"true",
        "mode":"server"
      }
    },
    "address-family":{
      "ipv4":{
        "enable":"true",
        "proxy-arp-enable": "true",
        "pool-name":"ipoe",
        "primary-dns":"198.51.100.1,
        "secondary-dns":"198.51.100.4"
      },
      "ipv6":{
        "enable":"true",
        "pool-name":"ipoe-ia-na",
        "prefix-delegation-pool-name":"ipoe-ia-pd",
```

```
        "primary-dns": "2001:db8:0:100::",
        "secondary-dns": "2001:db8:0:104::"
      }
    }
  }
}
```

### DHCP Server Configuration

DHCPv4 and DHCPv6 server configuration allows you assign IP addresses and manage subscriber settings for IPv4 and IPv6 clients. These settings control how access devices communicate with external DHCP servers, including how to reach the server, which source address to use.

#### Configuring DHCPv4 Server

You can configure the DHCP server for IP address allocation using the following command and options.

**Syntax:**

**set access dhcp-server** <server-name> **address** <address>
**set access dhcp-server** <server-name> **routing-instance** <routing-instance>
**set access dhcp-server** <server-name> **source-address** <source-address>

| Attribute | Description |
|---|---|
| `<cr>` | Without any option, the command configures a DHCPv6 server instance with the specified name. |
| `address` | Specifies the global DHCPv4 server IP address. |
| `routing-instance` | Defines the routing instance through which the DHCPv4 server is reachable. |
| `source-address` | Specifies the source IPv4 address used for DHCPv4 packets. |

The following example shows a DHCPv4 server configuration. It indicates the DHCP server name as 'dhcp-server1', which is the identifier for the DHCP server. The DHCP server's IP address is specified as '172.16.0.10', and the source IP address is specified as '172.16.0.2', which is used by the DHCP server when sending out packets. The routing instance name is set as 'default', which indicates that the DHCP server is accessible through the default instance.

```
supervisor@switch: cfg> show config access dhcp-server
{
  "rtbrick-config:dhcp-server": [
    {
      "server-name": "dhcp-server1",
      "address": "172.16.0.10",
      "source-address": "172.16.0.2",
      "routing-instance": "default"
```

```
    }
  ]
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Configuring DHCPv6 Server

The following commands and options are used to configure a DHCPv6 server.

**Syntax:**

**set access dhcpv6-server** <server-name> **address** <address>
**set access dhcpv6-server** <server-name> **routing-instance** <routing-instance>
**set access dhcpv6-server** <server-name> **source-address** <source-address>

| Attribute | Description |
|---|---|
| - | Without any option, the command creates a DHCPv6 server configuration. |
| address | Specifies the DHCPv6 server address. |
| routing-instance | Specifies the routing instance from which the DHCPv6 server is reachable. |
| source-address | Specifies the source address used for DHCPv6 packets. |

The following example shows a DHCPv6 configuration. It shows the DHCPv6 server name is specified as ‘dhcpv6-server1’ which is the identifier for the DHCPv6 server. The IP address is specified as '172:16::2'. The source IP address is specified as '172:16::10', which is used by the DHCPv6 server when sending out packets. The routing instance name is set as 'default'. It indicates the DHCPv6 server is accessible through the default instance.

```
supervisor@switch: cfg> show config access dhcpv6-server
{
  "rtbrick-config:dhcpv6-server": [
    {
      "server-name": "dhcpv6-server1",
      "address": "172:16::2",
      "source-address": "172:16::10",
      "routing-instance": "default"
    }
  ]
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

### AAA Profile Configuration

**Table:** `global.access.aaa.profile.config`

The AAA profile for subscriber access is a mandatory configuration that serves as a cornerstone for managing and tracking subscriber activities. It ensures that all subscribers are authenticated before accessing services, authorized to use those services, and accounted for billing and monitoring purposes.

The following diagram illustrates how the AAA profile is associated with various subscriber management tasks.

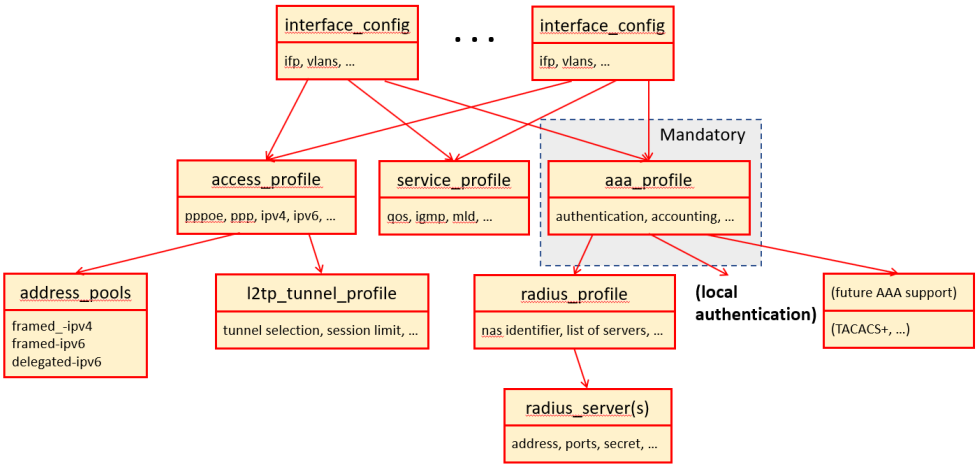



*Figure 68. AAA Profile Configuration*

#### Configuring the AAA Profile

The following command and options configure an AAA profile, which defines authentication, authorization, and accounting parameters for subscriber access.

**Syntax:**

```
set access aaa-profile <profile-name> aaa-radius-profile <aaa-radius-profile>
set access aaa-profile <profile-name> accounting <options>
set access aaa-profile <profile-name> authentication
set access aaa-profile <profile-name> idle-timeout <idle-timeout>
set access aaa-profile <profile-name> session-timeout <session-timeout>
```

| Attribute | Description |
|---|---|
| `aaa-profile <profile-name>` | Name of the AAA profile. |
| `accounting` | Configures accounting parameters for subscriber sessions. |
| `authentication` | Configures authentication parameters for subscriber access. |

| Attribute | Description |
| --- | --- |
| `session-timeout <session-timeout>` | The session timeout defines the maximum (uptime) duration, in seconds, that a subscriber can remain connected before the session is automatically terminated. The value '0' indicates that the session has no time limit and can continue indefinitely. Default: 0, Range: 0 - 4294967295. |
| `idle-timeout <idle-timeout>` | The idle timeout is the time, in seconds, after which a subscriber will be disconnected if no data is being transmitted through their outgoing logical interface (IFL). Control traffic is not included in these statistics. The subscriber is not considered idle as long as outgoing traffic is detected. The idle timeout is not limited but should be set to at least double the time of the logical interface statistics counter update interval (between 5 to 45 seconds). The idle timeout does not have a specific limit, but it is recommended to set it to at least double the time of the logical interface statistics counter update interval (which is between 5 to 45 seconds). Therefore, it is advisable to set a minimum idle timeout of 90 seconds. The value '0' means that the timeout is infinite Default: 0, Range: 0 - 4294967295 |
| `aaa-radius-profile` | The RADIUS profile, (for more details, see RADIUS Profile Configuration), which is used if RADIUS authentication or accounting is enabled. |

API To access the RESTCONF API that corresponds to this CLI, click include::common:partial$restconf_api_links.adoc[tag=set-access-aaa-profile].

Example:

This following example presents a AAA (Authentication, Authorization, and Accounting) profile configuration for RADIUS authentication and accounting. The profile name is 'aaa-radius.

The `session-timeout` value is set to '0', meaning there is no session timeout and the session can remain active indefinitely. The idle timeout value is also '0', indicating no idle timeout, so the session will not be terminated due to inactivity.

The parameter `aaa-radius-profile` is set to 'radius-default', which maps the AAA profile to the specific RADIUS profile named 'radius-default'. The RADIUS profile contains the necessary configuration details for communicating with the RADIUS server. In this profile, RADIUS is the selected method for handling all authentication requests. The `order` parameter indicates that RADIUS is the primary and only method of authentication configured.

For accounting, the order is specified as RADIUS, indicating that all accounting data is sent to the RADIUS server.

The ingress accounting-source is "POLICER", and the egress accounting source is "CLASS". "POLICER" indicates that the accounting data is based on a traffic policer, while "CLASS" indicates that the accounting data is based on traffic classes.

The `class-byte-adjustment-value` parameter is set to 16, adjusting the byte count for outgoing traffic by a specific value (16).

```
supervisor@switch: cfg> show config access aaa-profile aaa-radius
{
  "rtbrick-config:aaa-profile": {
    "profile-name": "aaa-radius",
    "session-timeout": 0,
    "idle-timeout": 0,
    "aaa-radius-profile": "radius-default",
    "authentication": {
      "order": "RADIUS"
    },
    "accounting": {
      "order": "RADIUS",
      "session-id-format": "DEFAULT",
      "ingress": {
        "accounting-source": "POLICER"
      },
      "egress": {
        "accounting-source": "CLASS",
        "class-byte-adjustment-value": 16
      }
    }
  }
}
```

The following example shows how to set the accounting metric key to `USERNAME` for a specific AAA profile.

```
supervisor@rtbrick.net: cfg> set access aaa-profile profile1 accounting metric-key
USERNAME
```

```
supervisor@rtbrick.net: cfg> show conf access access-profile
{
  "rtbrick-config:access-profile": [
    {
      "profile-name": "profile1",
      "accounting": {
        "metric-key": "USERNAME"
      }
    }
  ]
```

#### Configuring Authentication

RBFS supports the authentication methods: NONE, LOCAL, RADIUS and LOCAL_RADIUS.

- NONE: Disables authentication, accepting all credentials without verification.
- LOCAL: Authenticates the subscriber based on locally defined user profiles. For more details, refer to User Profile Configuration.
- RADIUS: Authenticates the subscriber remotely by sending an authentication request to the configured RADIUS servers.
- LOCAL_RADIUS: Authenticates the subscriber based on locally defined user profiles. If no matching local user is found, the system attempts authentication via RADIUS. If a local user is found but the password does not match, the subscriber is immediately rejected, and no RADIUS request is made.

**Syntax**:

> **set access aaa-profile** <profile-name> **authentication order** [NONE | LOCAL | RADIUS | LOCAL_RADIUS]

The following command and options allow you to define the order of authentication methods.

```
set access aaa-profile aaa-profile1 authentication order LOCAL_RADIUS
```

| Attribute | Description |
|---|---|
| `order` <NONE \| LOCAL \| RADIUS \| LOCAL_RADIUS> | This option defines the order of authentication methods. The options include NONE, LOCAL, RADIUS, and LOCAL_RADIUS. Default: NONE. |

```
supervisor@rtbrick.net: cfg> show config access aaa-profile aaa-profile1
authentication
{
  "rtbrick-config:authentication": {
    "order": "RADIUS"
  }
}
```

API To access the RESTCONF API that corresponds to this CLI, click include::common:partial$restconf_api_links.adoc[tag=set-access-aaa-profile-authentication].

#### Configuring Accounting

Subscriber accounting involves tracking and recording a subscriber session duration and data usage. This process includes two main components:

- Time Accounting: Measures the duration of a subscriber's session.
- Volume Accounting: Tracks the number of packets and bytes transmitted or received.

Volume accounting operates in both directions (ingress and egress), and these can be configured

independently.

Currently, RBFS supports RADIUS as the only accounting method.

**Syntax**:

**set access aaa-profile** <profile-name> **accounting**
**set access aaa-profile** <profile-name> **accounting egress** <option>
**set access aaa-profile** <profile-name[> **accounting ingress** <option>

The following sections provide information about Ingress Accounting, Egress Accounting, and Accounting Adjustments.

| Attribute | Description |
|---|---|
| `-` | Without any option, it sets accounting for the AAA profile. |
| `egress` | Egress volume accounting. |
| `ingress` | Ingress volume accounting option |

```
supervisor@rtbrick.net: cfg> show config access aaa-profile aaa-profile1 accounting
{
  "rtbrick-config:accounting": {
    "order": "RADIUS",
    "interim-interval": 60,
    "session-id-format": "DEFAULT"
  }
}
```

To access the RESTCONF API that corresponds to this CLI, click include::common:partial$restconf_api_links.adoc[tag=set-access-aaa-profile-accounting].

#### Configuring Accounting Adjustments

The accounting adjustment feature enables basic counter modifications for the configured accounting method, such as RADIUS accounting. This configuration is necessary to normalize counters across different platforms in each direction. On Broadcom Q2C and Q2A based platforms, packets are counted in the size they enter the switch. Without adjustment, egress accounting would count downstream traffic as received from the core, complete with MPLS labels, while ingress accounting typically includes VLAN headers and/or PPPoE headers.

This counter adjustment aims to normalize counters with diverse encapsulations (double-tagged, untagged, and so on), potentially aligning to L3 counters (IP header and payload) as an example, or exclusively adapting egress traffic to match the outgoing packet encapsulation. The possibility for separate adjustment configurations per direction allows parity in the counters for both ingress and egress.

In a scenario where packets enter a device with various encapsulations like PPPoE headers, MPLS

labels, or VLAN tags, the system may count these additional headers in the packet size if the adjustment feature is not enabled. This can cause discrepancies in accounting reports.

For instance, an egress interface where traffic exits the device and goes to subscribers. If this traffic originally came from the core network with MPLS labels attached, the system counts the bytes of those labels in its accounting, inflating the byte count. Similarly, on the ingress side, the device counts VLAN tags or PPPoE headers, further impacting accurate accounting reports. To address this issue, you can configure a byte adjustment on both the ingress and egress sides to normalize these counters.

In RBFS, two configuration options are available for to achieve this: the `byte adjustment value` and the `byte adjustment factor`. The option `byte adjustment value`, which can be positive or negative (for example, -20 or 20), adjusts the byte count by a specific amount. Any decimal digits in the adjustment value are rounded to the nearest whole number (for example, 20.2 becomes 20). The option `byte adjustment factor`, which is less commonly used, adjusts the byte count by a percentage. It accepts positive values and considers up to two decimal places, such as 0.98 for a 2% reduction or 1.02 for a 2% increase.

You can configure accounting adjustments for Ingress Accounting and Egress Accounting as described in the following sections.

#### Ingress Accounting

The ingress accounting configuration defines how incoming traffic counters are collected and adjusted for AAA accounting for the specified AAA profile. Subscriber ingress accounting refers to the process of measuring and recording the data usage or traffic that enters a subscriber interface (upstream).

**Syntax:**

**set access aaa-profile** <profile-name> **accounting ingress accounting-source** [**IFL** | **POLICER**]
**set access aaa-profile** <profile-name> **accounting ingress byte-adjustment-factor** [0.00 - 2.00]
**set access aaa-profile** <profile-name> **accounting ingress byte-adjustment-value** [-32 - 32]
**set access aaa-profile** <profile-name> **accounting ingress policer-byte-adjustment-factor** [0.00 - 2.00]
**set access aaa-profile** <profile-name> **accounting ingress policer-byte-adjustment-value** [-32 - 32]

The following command options allow you to set data usage accounting for incoming traffic.

| Attribute | Description |
|---|---|
| `accounting-source` | This option provides control over the counters used for subscriber ingress accounting when RADIUS accounting is enabled. The counters in question are the RADIUS attributes `Acct-Input-Packets` (47), `Acct-Input-Octets` (42), and `Acct-Input-Gigawords` (52).<br><br>By default, the policer statistics (POLICER) are utilized, which represent the total traffic accepted across all policer levels (1-4). However, ingress control traffic is subject to a separate control plane policer and is therefore not included in the session policer statistics. Consequently, policers are necessary if session accounting is required.<br><br>Alternatively, the logical interface (IFL) statistics can be used, which capture all received traffic, including control traffic and the traffic that is dropped by the ingress policer. It is important to note that this option may NOT be available on all platforms. Default: POLICER, Supported values: POLICER and IFL. |
| `byte-adjustment-value` | Adjusts ingress IFL counters by +/- N bytes per packet. It allows you to adjust the ingress IFL counters by adding or subtracting a specified number of bytes per packet. Default: 0.00, Range: -32 to 32 |
| `byte-adjustment-factor` | Adjust the ingress IFL counters by applying a specified factor. (This adjustment is made after the `byte-adjustment-value` is applied). Default: 1.00 Range: 0.00 - 2.00. |
| `policer-byte-adjustment-value` | Adjusts ingress POLICER counters by +/- N bytes per packet. It adjusts the ingress POLICER counters by adding or subtracting a specified number of bytes per packet. Default: 0.00 Range: -32 to 32 |
| `policer-byte-adjustment-factor` | Adjusts the ingress POLICER counters by applying a specified factor. This adjustment is also executed after the `policer-byte-adjustment-value` is applied. Default: 1.00 Range: 0.00 - 2.00. |

#### Egress Accounting

Subscriber egress accounting refers to the process of measuring and recording the data usage or traffic that is sent from a subscriber interface (downstream).

**Syntax:**

**set access aaa-profile** <profile-name> **accounting egress accounting-source** <*IFL* | **CLASS**>
**set access aaa-profile** <profile-name> **accounting egress byte-adjustment-factor** <0.00 - 2.00>
**set access aaa-profile** <profile-name> **accounting egress byte-adjustment-value** [-32 - 32]
**set access aaa-profile** <profile-name> **accounting egress class-byte-adjustment-factor** [0.00 - 2.00]
**set access aaa-profile** <profile-name> **accounting egress class-byte-adjustment-value** [-32 -

```
32]
```

For the command options and description, refer to the preceding Ingress Accounting Configuration section.

### RADIUS Profile Configuration

Subscriber management allows the configuration of a RADIUS profile, which is mandatory if RADIUS is used for authentication or accounting.

The diagram below illustrates how the RADIUS profile is associated with the subscriber management tasks at a broader level.

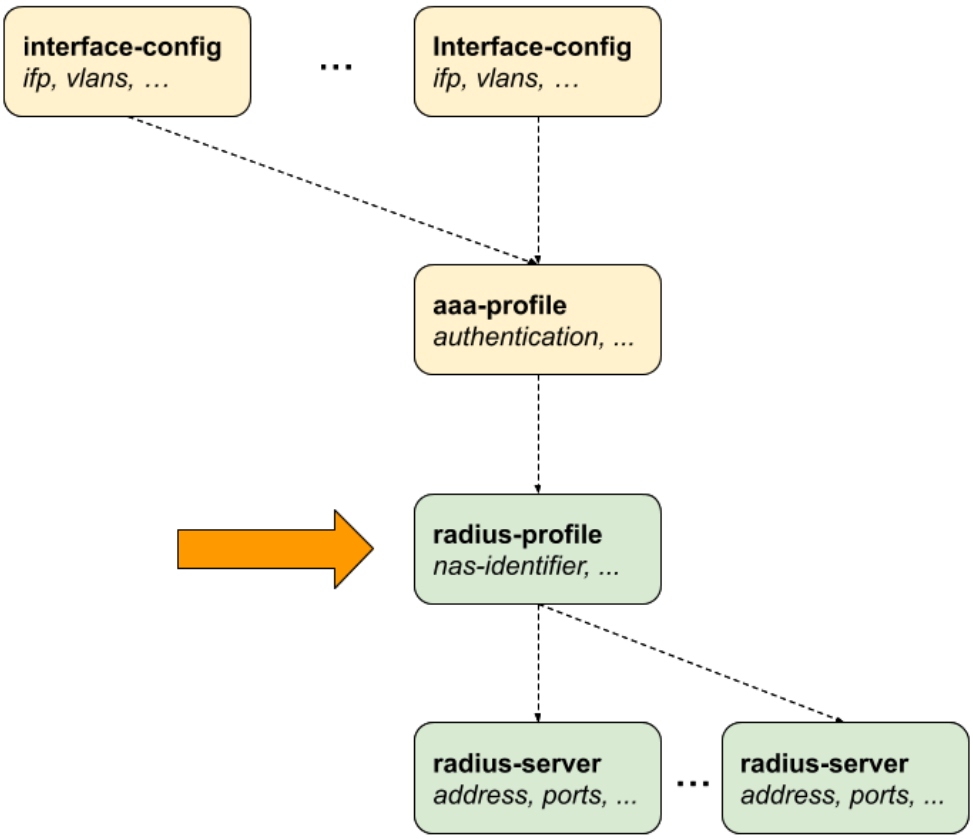



*Figure 69. RADIUS Profile Configuration*

#### Configuring the RADIUS Profile

The RADIUS profile configuration involves setting up various parameters that define how the router interacts with the RADIUS server for authentication, accounting, and so on.

```
set access radius-profile <profile-name> accounting accounting-backup [false | true]
set access radius-profile <profile-name> accounting accounting-backup-max [1 - 4294967295]
set access radius-profile <profile-name> accounting accounting-on-off [false | true]
set access radius-profile <profile-name> accounting accounting-on-wait [false | true]
set access radius-profile <profile-name> accounting algorithm-type [DIRECT | ROUND-ROBIN]
```

**set access radius-profile** <profile-name> **accounting radius-server-profile-name** <radius-server-profile-name>
**set access radius-profile** <profile-name> **accounting stop-on-reject [false | true]**
**set access radius-profile** <profile-name> **authentication algorithm-type [DIRECT | ROUND-ROBIN]**
**set access radius-profile** <profile-name> **authentication radius-server-profile-name** <radius-server-profile-name>
**set access radius-profile** <profile-name> **calling-station-id-format [NONE | MAC]**
**set access radius-profile** <profile-name> **nas-identifier** <nas-identifier>
**set access radius-profile** <profile-name> **nas-ip-address** <nas-ip-address>
**set access radius-profile** <profile-name> **nas-port-format [DEFAULT | SLOTS | PORTS]**
**set access radius-profile** <profile-name> **nas-port-type [Virtual | SDSL | ADSL-CAP | ADSL-DMT | Ethernet | xDSL | Cable | PPPoA | PPPoEoA | PPPoEoE | PPPoEoVLAN | PPPoEoQinQ | xPON]**

| Attribute | Description |
|---|---|
| `calling-station-id-format` | Set the value for RADIUS attribute Calling-Station-Id (31). The default setting is NONE, meaning the attribute is not sent. If set to MAC, the client MAC address is sent as a string. Default: NONE. |
| `nas-identifier` | Set the value for the RADIUS attribute NAS-Identifier (32). Default: system hostname. |
| `nas-ip-address` | Set the value for RADIUS attribute NAS-IP-Address (4). Default: source IPv4 address. |
| `nas-port-type` | Set the value for RADIUS attribute NAS-Port-Type (61). Default: Ethernet. |
| `nas-port-format` | Set the format of the 32-bit RADIUS attribute NAS-Port (5). |

| Name | Bits | Values |
|---|---|---|
| DEFAULT | 1:1:6:12:12 | slot:subslot:port:vlan:vlan |
| SLOTS | 6:2:6:12:6 | slot:subslot:port:vlan:vlan |
| PORTS | 0:1:7:12:12 | slot:subslot:port:vlan:vlan |

API To access the RESTCONF API that corresponds to this CLI, click here.

The following example shows a typical RADIUS profile for authentication and accounting. The RADIUS profile name is specified as 'radius-default'. NAS Identifier is set as BNG and NAS Port Type is specified as 'Ethernet'. RADIUS servers used for authentication is set 'radius-server-1' and 'radius-server-2'. RADIUS servers used for accounting is also set as 'radius-server-1' and 'radius-server-2'.

The Accounting RADIUS servers are specified as 'radius-server-1' 'radius-server-2'. For information about other options, see the table descriptions below.

```
supervisor@switch: cfg> show config access radius-profile radius-default
{
  "rtbrick-config:radius-profile": {
```

```
    "profile-name": "radius-default",
    "nas-identifier": "BNG",
    "nas-port-type": "Ethernet",
    "authentication": {
      "radius-server-profile-name": [
        "radius-server-1",
        "radius-server-2"
        ]
    },
    "accounting": {
      "radius-server-profile-name": [
        "radius-server-1",
        "radius-server-2"
        ],
      "stop-on-reject": "true",
      "stop-on-failure": "true",
      "accounting-on-off": "true",
      "accounting-on-wait": "true",
      "accounting-backup": "true",
      "accounting-backup-max": 86400
    }
  }
}
```

**Configuring Authentication**

The following configuration command and options set the 'radius-profile' as authentication method. It also defines the list of RADIUS servers and the algorithm for RADIUS high-availability.

| Attribute | Description |
|---|---|
| - | Without any option, the command sets authentication for the RADIUS profile. |
| radius-server-profile-name | List of multiple RADIUS servers used for authentication. |
| algorithm-type | Specifies the authentication server selection algorithm. For more information, see RADIUS Redundancy. Default: DIRECT Values: DIRECT, ROUND-ROBIN |

```
supervisor@rtbrick.net: cfg> show config access radius-profile aaa-radius1
authentication radius
-server-profile-name
{
  "rtbrick-config:radius-server-profile-name": [
    "radius-srv1",
    "radius-srv2"
    ]
}
```

To access the RESTCONF API that corresponds to this CLI, click here.

#### Configuring Accounting

Configuring accounting involves specifying its name and defining various parameters. The following command and options allow you to configure RADIUS accounting.

| Attribute | Description |
|---|---|
| `-` | Without any option, the command sets accounting for the RADIUS profile. |
| `radius-server-profile-name` | List of RADIUS servers used for accounting. |
| `algorithm-type` | Specifies the accounting server selection algorithm. For more information, see RADIUS Redundancy. Default: DIRECT, Values: DIRECT and ROUND-ROBIN. |
| `stop-on-failure` | If set to `true`, the accounting will stop, if there is a failure in the process after authentication was accepted. Default: False. |
| `stop-on-reject` | If set to `true`, when the authentication is rejected, the accounting process will stop. Default: False. |
| `accounting-on-off` | Enables RADIUS Accounting-On/Off messages. For more information, see RADIUS Accounting. Default: False. |
| `accounting-on-wait` | Waits for an `Accounting-On` response ensuring that no new subscriber is allowed until the accounting process has been initiated. Default: False. |
| `accounting-backup` | Enables backup for accounting (optional). RADIUS accounting requests are often used for billing and, therefore should be able to store and retry over a longer period (commonly, up to 24 hours or more). Default: False. |
| `accounting-backup-max` | If enabled, this option defines maximum backup accounting hold time, in seconds. Default: 3600, Range: 1 - 4294967295. |

```
supervisor@DT-STD-23-2401>bm13-tst.htz.rtbrick.net: cfg> show config access radius-
profile aaa-radius1 accounting
{
  "rtbrick-config:accounting": {
    "radius-server-profile-name": [
      "radius-srv1",
      "radius-srv2"
      ]
  }
}
```

To access the RESTCONF API that corresponds to this CLI, click here.

### RADIUS Server Configuration

Subscriber Management often relies on the RADIUS server for authentication, authorization, and accounting. There are alternative AAA solutions, including local methods that operate

independently of network availability.

The RADIUS server configuration is not standalone; it is inherently dependent on the configuration of a RADIUS profile. You must first create a RADIUS profile before configuring the associated RADIUS server.

The following diagram illustrates how RADIUS server configuration fits into the broader context of subscriber management tasks.

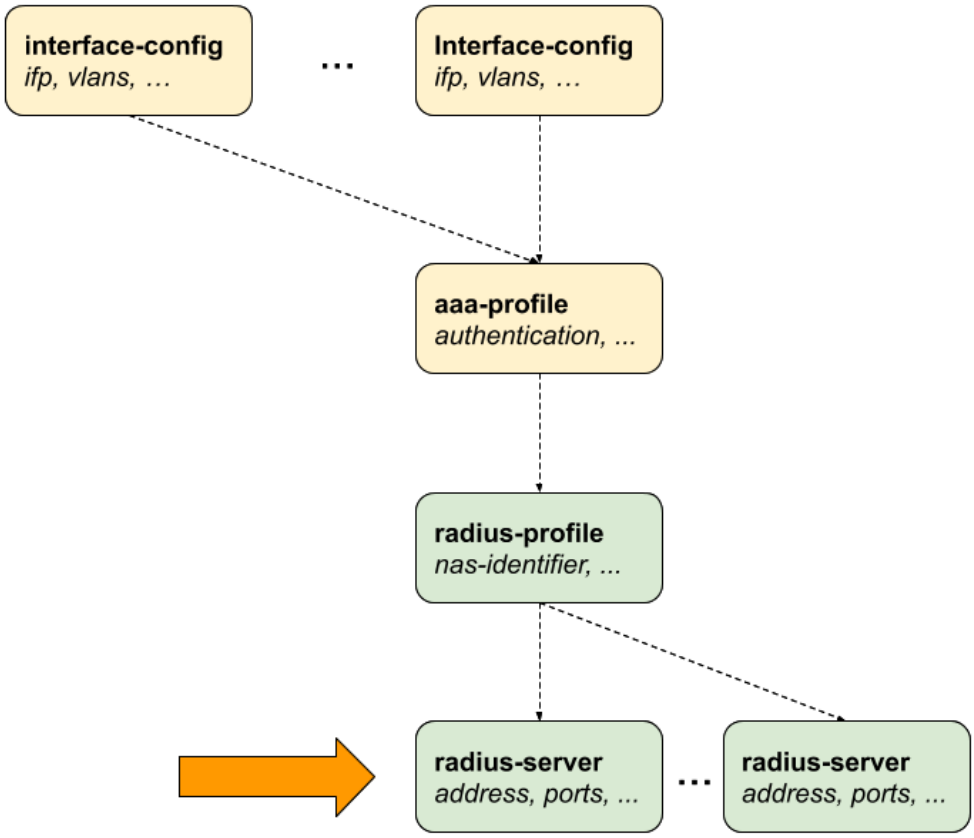



*Figure 70. RADIUS Server Configuration*

#### Configuring the RADIUS Server

The following configuration command and options are used to configure a RADIUS server.

```
set access radius-server <server-name> address <address>
set access radius-server <server-name> authentication enable [false | true]
set access radius-server <server-name> authentication outstanding [1 - 65535]
set access radius-server <server-name> authentication port [1 - 65535]
set access radius-server <server-name> authentication retry [1 - 255]
set access radius-server <server-name> authentication timeout [1 - 65535]
set access radius-server <server-name> coa enable [false | true]
set access radius-server <server-name> coa port [1 - 65535]
set access radius-server <server-name> message-authenticator [false | true]
set access radius-server <server-name> rate [1 - 65535]
set access radius-server <server-name> routing-instance <routing-instance>
set access radius-server <server-name> secret-encrypted-text <secret-encrypted-text>
set access radius-server <server-name> secret-plain-text <secret-plain-text>
```

**set access radius-server** <server-name> **source-address** <source-address>

| Attribute | Description |
|---|---|
| `<server-name>` | Specifies the name used to identify the RADIUS server within the system. |
| `accounting` | Enables and configures RADIUS accounting, allowing the system to send session usage and lifecycle information (start, stop, interim updates) to the server. |
| `address` | Defines the IP address or hostname of the RADIUS server to which authentication and accounting requests are sent. NOTE: Multiple RADIUS servers with the same IPv4 address are not supported, even if the instance or port is different. |
| `authentication` | Enables and configures RADIUS-based authentication for validating subscriber or user credentials. |
| `coa` | Enables support for Change of Authorization (CoA), allowing the RADIUS server to dynamically modify session parameters after authentication. |
| `message-authenticator true \| false` | Enables or disables the requirement for the Message-Authenticator attribute in Access-Response messages from the specified RADIUS server. True indicates that the Message-Authenticator attribute is required, while False indicates that it is not required. Default: False. |
| `rate` | Specifies the maximum number of RADIUS requests that can be sent to the server per second, used to control traffic and prevent server overload. Default: 600. Range: 1 - 65535. |
| `routing-instance` | Associates the RADIUS server communication with a specific routing instance (VRF), enabling reachability in multi-VRF deployments. |
| `secret-encrypted-text` | Configures the shared secret key in encrypted form for secure communication between the device and the RADIUS server. |
| `secret-plain-text` | Configures the shared secret key in plain text format; typically used for initial setup before encryption is applied. |
| `source-address` | Specifies the source IPv4 address used for outgoing RADIUS packets, ensuring consistent identification and proper routing. |

API To access the RESTCONF API that corresponds to this CLI, click here.

The following example shows the RADIUS server configuration for authentication and accounting. Each of these options allows you to define how the RADIUS server will operate. The command sets RADIUS server name as 'radius-server-1'. The configuration specifies the IP address '198.51.100.101' of the RADIUS server, which the router will use to send RADIUS requests. The source IP address is specified as '198.51.100.200'. It sets 'CoA' (Change-of-Authorization), a feature that allows the RADIUS server to change the authorization parameters of a session after it has been established.

The `secret-encrypted-text` parameter is specified that configures the RADIUS secret in an encrypted format. The key is used for encrypting RADIUS messages between the client and server to ensure security.

```
supervisor@switch: cfg> show config access radius-server radius-server-1
{
  "rtbrick-config:radius-server": {
    "server-name": "radius-server-1",
    "address": "198.51.100.101",
    "source-address": "198.51.100.200",
    "secret-encrypted-text": "$21e4946e31b406de98b3077aef03ed5a7",
    "authentication": {
      "enable": "true"
    },
    "accounting": {
      "enable": "true"
    },
    "coa": {
      "enable": "true"
    }
  }
}
```

#### Configuring Authentication

The following command and options allow to configure the authentication settings on a RADIUS server. The `enable` option is used to activate RADIUS authentication for the specified RADIUS server 'radius-server-1'. The `outstanding` option allows you to specify the maximum number of pending authentication requests that the device can have at any given time. The `port` option enables you to define the port number used for RADIUS authentication requests. The `retry` option defines the maximum number of retries the device makes if the RADIUS server does not acknowledge an authentication request. This ensures that transient network issues do not cause authentication failures. The `timeout` setting determines the time, in seconds, that the device will wait for a response from the RADIUS server after sending authentication requests. If the server does not respond within this time frame, the request will be considered timed out.

**set access radius-server** <server-name> **authentication enable** [**false** | **true**]
**set access radius-server** <server-name> **authentication outstanding** [1 - 65535]
**set access radius-server** <server-name> **authentication port** [1 - 65535]
**set access radius-server** <server-name> **authentication retry** [1 - 255]
**set access radius-server** <server-name> **authentication timeout** [1 - 65535]

| Attribute | Description |
|---|---|
| `-` | Without any option, the command sets RADIUS server authentication. |
| `enable` | Enables RADIUS authentication. Default: False. |
| `port` | RADIUS authentication port. Default: 1812. Range: 1 - 65535. |

| Attribute | Description |
|---|---|
| `retry` | This option specifies the number of authentication retries before declaring this server as unreachable for authentication. After reaching the limit, the client begins to send requests to other RADIUS servers and rejects the request after receiving the end of the list. Default: 3. Range: 1 - 255. |
| `timeout` | Authentication request timeout, in seconds. Default: 5. Range: 1 - 65535. |
| `outstanding` | This option specifies the maximum number of outstanding authentication requests for this RADIUS server. A request is counted as outstanding if sent out but the response is not received. Default: 100. Range: 1 - 65535. |

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Configuring Accounting

RADIUS accounting allows the tracking of user sessions. This configuration sets up RADIUS accounting on a specified RADIUS server. When enabled, the device will send accounting data to the RADIUS server.

**Syntax**:

**set access radius-server** <server-name> **accounting enable** [**false** | **true**]
**set access radius-server** <server-name> **accounting outstanding** [1 - 65535]
**set access radius-server** <server-name> **accounting port** [1 - 65535]
**set access radius-server** <server-name> **accounting retry** [1 - 255]
**set access radius-server** <server-name> **accounting timeout** [1 - 65535]

| Attribute | Description |
|---|---|
| `enable` | Enable RADIUS accounting. Default: False. |
| `port` | This specifies the port number on the RADIUS server that is used for accounting purpose. Default: 1813. Range: 1 - 65535. |
| `retry` | This option determines the maximum number of attempts the device will make to send an accounting request to the RADIUS server before marking the server as unreachable for accounting. Once the retry limit is reached, the device will attempt to send accounting requests to other configured RADIUS servers. Default: 10. Range: 1 - 255. |
| `timeout` | Accounting request timeout, in seconds. Default: 30. Range: 1 - 65535. |
| `outstanding` | This option specifies the maximum number of outstanding accounting requests for this RADIUS server. A request is counted as outstanding if sent out, but the response is not received. Default: 100. Range: 1 - 65535. |

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Configuring Change-of-Authorization (CoA)

Change-of-Authorization (CoA) allows a RADIUS server to send requests to the network device to change the authorization parameters of an active session. The following configuration involves setting up CoA for a specified RADIUS server called 'radius-server-1'.

**Syntax**:

```
set access radius-server <server-name> coa enable
set access radius-server <server-name> coa port [1 - 65535]
```

| Attribute | Description |
|---|---|
| `enable` | This option enables the device to receive CoA requests from the specified RADIUS server. Default: False. |
| `port` | Specifies the device port that listens/receives for CoA requests from the RADIUS server. Default: 3799. Range: 1 - 65535. |

```
supervisor@rtbrick.net: cfg> show config access radius-server radius-srv2 coa
{
  "rtbrick-config:coa": {
    "enable": "true"
  }
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Service Profile Configuration

You can set up Service Profile to assign specific Quality of Service (QoS) and IGMP settings to individual subscribers. This ensures that each subscriber gets the right level of service based on their requirements.

The Service Profile configuration is optional.

The following diagram illustrates how the service profile is associated with other subscriber management tasks.

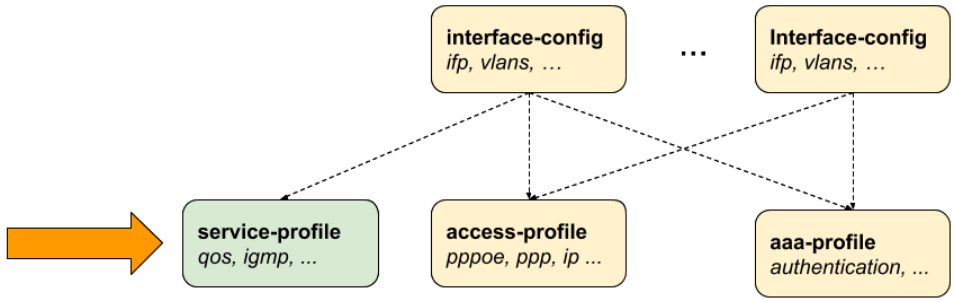



*Figure 71. Service Profile Configuration*

#### Configuring the Service Profile

The following command and options are used to set up a service profile named 'iptv'.

**set access service-profile** <profile-name> **acl**
**set access service-profile** <profile-name> **address-translation**
**set access service-profile** <profile-name> **http-redirect**
**set access service-profile** <profile-name> **igmp**
**set access service-profile** <profile-name> **qos**

| Attribute | Description |
|---|---|
| `-` | Without any option, the command creates the specified service profile. |
| `<profile-name>` | Specifies the name of the service profile. |
| `acl` | Configures ACL-related attributes for the service profile. |
| `address-translation` | Configures CGNAT-related attributes for the service profile. |
| `http-redirect` | Configures HTTP redirect-related attributes for the service profile. |
| `igmp` | Configures IGMP-related attributes for the service profile. |
| `qos` | Configures QoS-related attributes for the service profile. |

The following example shows a service profile configuration for subscribers with IPTV (multicast) services. The configuration defines specific QoS and IGMP settings that will be applied to subscribers using this service profile. In this configuration, the service profile named ‘iptv’ is linked to a QoS profile named ‘iptv-qos-xl’. This QoS profile defines the prioritization and bandwidth allocation for IPTV traffic for the subscriber profile that it is associated with.

Additionally, the configuration specifies the IGMP settings. Enabling IGMP is achieved by setting it to 'true' for the service profile that enables subscribers to join multicast groups. The configuration specifies the IGMP profile named 'iptv-basic', with IGMP version 3 in use. The parameter `max-members` is set to 10, which limits the number of simultaneous multicast groups (maximum IGMP memberships) that a subscriber can join.

```
supervisor@switch: cfg> show config access service-profile iptv
{
  "rtbrick-config:service-profile": {
    "profile-name": "iptv",
    "qos": {
      "profile": "iptv-qos-xl"
    },
    "igmp": {
      "enable": "true",
      "profile": "iptv-basic",
      "version": "IGMPv3",
      "max-members": 10
    }
  }
```

```
}
```

To access the RESTCONF API that corresponds to this CLI, click here.

#### Configuring QoS

The following QoS configuration defines the QoS settings for the service profile 'iptv'.

| Attribute | Description |
|---|---|
| `parent-scheduler` | This option defines the parent scheduler element within the scheduler-map that is assigned to the subscriber. If the parent scheduler is not specified, the scheduler-map will be directly bound to the local IFP where the session is established.<br><br>This attribute can only be set once and cannot be changed without disconnecting the session. The parent scheduler can also be configured through RADIUS, which gets priority over the local configuration.<br><br>Assigning a QoS parent scheduler that does not exist on the corresponding IFP will result in blackholing all egress data traffic. Control traffic will remain unaffected, allowing the session to persist despite the data loss. |
| `profile` | This option assigns a specific QoS configuration profile to the subscriber. The QoS profile can be also set through RADIUS. If configured both locally and through RADIUS, the RADIUS setting gets priority. |

To access the RESTCONF API that corresponds to this CLI, click here.

#### Configuring IGMP

This configuration sets IGMP options in the service profile to provide multicast IPTV to subscribers. It lets you control IGMP features like turning the service on, limiting group memberships per subscriber, linking an IGMP profile, and choosing the protocol version. These settings help ensure efficient multicast, scalability, and support for different client types.

The following IGMP configuration defines the settings for the service profile 'iptv'.

**set access service-profile** <profile-name> **igmp enable [false** | **true]**
**set access service-profile** <profile-name> **igmp max-members** <max-members>
**set access service-profile** <profile-name> **igmp profile** <profile>
**set access service-profile** <profile-name> **igmp version [IGMPv1** | **IGMPv2** | **IGMPv3]**

| Attribute | Description |
|---|---|
| `enable` | This option dynamically enables or disables IGMP for a subscriber. Default: False. |

| Attribute | Description |
|---|---|
| `max-members` | This option limits the number of simultaneous multicast channels (maximum IGMP memberships) a subscriber can join. Default: 1. Range: 1 - 4294967295. |
| `profile` | This option specifies the IGMP profile to be associated with the subscriber. |
| `version` | This defines the version of IGMP for a subscriber. Default: V3. Values: V1, V2, and V3. |

API To access the RESTCONF API that corresponds to this CLI, click here.

### L2TP Profile Configuration

The L2TPv2 profile specifies the parameters needed to establish and manage L2TP tunnels between the access node (LAC) and the L2TP Network Server (LNS). Through this profile, you can control key attributes such as tunnel endpoints, authentication methods, and session management.

Configuring an L2TP profile is optional in subscriber management and is only required when L2TP-based tunneling is used for subscriber sessions. If L2TP is not part of the deployment, subscriber sessions can be handled without this profile.

The following diagram illustrates how the L2TP profile configuration is are associated with the other Subscriber Management tasks.

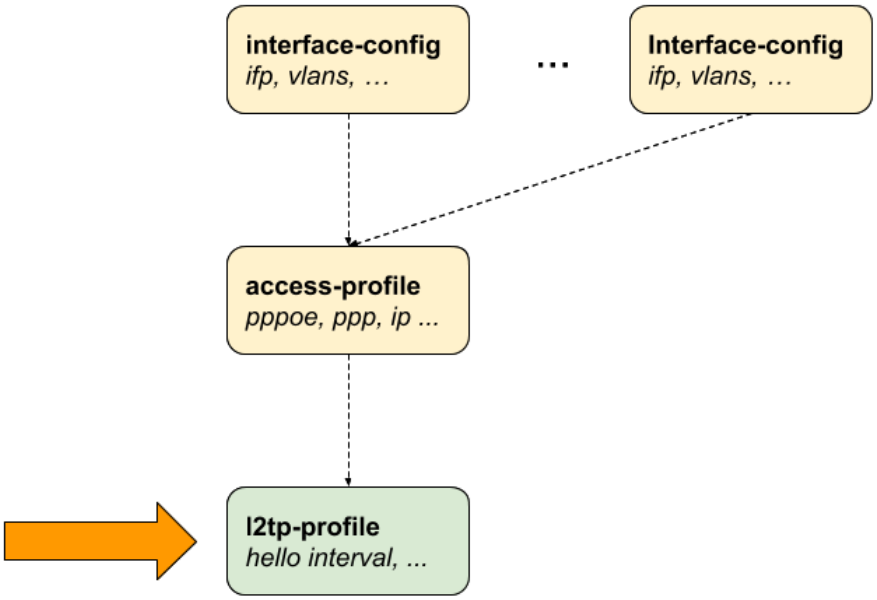



*Figure 72. L2TPv2 Profile Configuration*

#### Configuring the L2TP Profile

The following command and options allow you to configure an L2TP profile.

**Syntax**:

**set access l2tp-profile** <profile-name> **client-ipv4** <client-ipv4>
**set access l2tp-profile** <profile-name> **client-name** <client-name>
**set access l2tp-profile** <profile-name> **connect-speed-update [false | true]**
**set access l2tp-profile** <profile-name> **dead-timeout-interval** [1 - 4294966]
**set access l2tp-profile** <profile-name> **hello-interval** [0 - 86400]
**set access l2tp-profile** <profile-name> **hide-authentication [false | true]**
**set access l2tp-profile** <profile-name> **idle-timeout-interval** [0 - 4294966]
**set access l2tp-profile** <profile-name> **inactive-timeout-interval** [1 - 4294966]
**set access l2tp-profile** <profile-name> **instance** <instance>
**set access l2tp-profile** <profile-name> **pon-access-line-version [DISABLED | DRAFT-LIHAWI-00 | DRAFT-LIHAWI-04]**
**set access l2tp-profile** <profile-name> **pool-name** <pool-name>
**set access l2tp-profile** <profile-name> **receive-window** [1 - 256]
**set access l2tp-profile** <profile-name> **request-retries** [1 - 600]
**set access l2tp-profile** <profile-name> **request-timeout-interval** [1 - 30]
**set access l2tp-profile** <profile-name> **retransmit-interval** [1 - 30]
**set access l2tp-profile** <profile-name> **selection-algorithm [BALANCED | RANDOM]**
**set access l2tp-profile** <profile-name> **service-label** <service-label>
**set access l2tp-profile** <profile-name> **session-limit** [1 - 65535]

| Attribute | Description |
|---|---|
| `client-ipv4` | This is the default IPv4 address for the local L2TP tunnel client (LAC), if no specific address is provided through the L2TP pool or RADIUS configuration. |
| `client-name` | This is the default hostname for the local L2TP tunnel client (LAC) if no specific hostname is provided through the L2TP pool or RADIUS configuration. Default: system hostname. |
| `instance` | The routing instance in which the L2TP endpoint (LNS) is reachable. Default: Default. |
| `service-label` | Define the service label to support L2TP over MPLS. For more details, see Configuring L2TP over MPLS.<br><br>For details on the supported label ranges, see the Static, System, and Dynamic Label Ranges section in the Platform Guide. |
| `selection-algorithm` | It defines how to select a tunnel from a pool of available LNS servers. For more information, see L2TP Tunnel Selection.<br><br>The RANDOM algorithm selects a tunnel randomly, while the BALANCED algorithm chooses the tunnel with the fewest sessions. Default: BALANCED, Supported values: BALANCED and RANDOM. |
| `session-limit` | This is the default session limit for a tunnel unless otherwise specified. Tunnels that have reached their session limit are not considered for additional sessions. Default: 64000, Range: 1 - 65535. |

| Attribute | Description |
|---|---|
| `pool-name` | This allows to assign a default L2TP tunnel pool. For more information, see L2TP Tunnel Pool Configuration. This default can be overwritten by user-defined pool names from the local user profiles (For more information, see User Profile Configuration) or received through RADIUS attribute RtBrick-L2TP-Pool (VSA 26-50058-40). |
| `hello-interval` | Specifies the L2TP tunnel hello interval time, in seconds. '0' indicates disabled.<br><br>The HELLO keep-alive messages are part of the L2TP control channel. For more information, see L2TP Control Channel. These messages are sent only when the message queue is empty, and no other messages have been transmitted during the hello interval. Default: 30, Range: 0 - 86400. |
| `idle-timeout-interval` | This interval defines the maximum time, in seconds, to keep a tunnel without sessions established. The session will remain forever if this value is set to 0. Default: 600, Range: 0 - 4294966. |
| `dead-timeout-interval` | This interval defines the time, in seconds, This interval defines the time, in seconds, that a tunnel remains in the DEAD state if it becomes unreachable. Once the interval expires, the tunnel transitions back to the DOWN state, making it available for new sessions. Default: 300, Range: 1 - 4294966. |
| `inactive-timeout-interval` | This interval defines the time, in seconds, to keep an inactive tunnel before removal. This interval is reset with every new session request which considers this tunnel as a potential candidate. Default: 900, Range: 1 - 4294966. |
| `receive-window` | Specifies the receive window size offered to the remote peer trough the Receive Window Size AVP (10) in SCCRQ or SCCRP message.<br><br>For example, if a receive window size of '8' is advertised in the SCCRQ or SCCRP message, the remote peer can send up to 8 unacknowledged control messages. After reaching this limit, it must wait for an acknowledgment to advance the window before sending additional messages. Default:* 8, Range: 1 - 256. |
| `request-retries` | Defines the number of times the system attempts to establish a tunnel before considering it unsuccessful. See also the`request-timeout-interval` attribute. Default: 5, Range: 1 - 600. |
| `request-timeout-interval` | This interval, when multiplied by the number of request retries, determines the maximum time (in seconds) to wait for the selected tunnel to establish before selecting to another tunnel from the list. Default: 1, Range: 1 - 30.<br><br>Modify the values for `request-retries` and `request-timeout-interval` attributes with caution, as incorrect value may lead to delay in tunnel establishment. |

| Attribute | Description |
|---|---|
| `retransmit-interval` | This value specifies the interval, in seconds, between retransmissions of a message.<br><br>Each successive retransmission uses an exponential backoff interval, meaning the interval doubles after each retransmission. For example, if the first retransmission occurs after 1 second, the next will occur after 2 seconds, then 4 seconds, 8 seconds, 16 seconds, 32 seconds, and finally 64 seconds. After a maximum of 6 retransmissions, the interval reaches its maximum value. For a retransmit interval of 1, the total time can reach 64 seconds, and for an interval of 2, it can extend up to 128 seconds and so on. Default: 1, Range: 1 - 30. |
| `hide-authentication` | When enabled, the L2TP proxy will hide the authentication response AVP if the authentication type is PAP, ensuring that the password is not transmitted in plain text. Default: False. |
| `pon-access-line-version` | Adding additional PON attributes to the L2TP access line information. (For more information, see L2TP Access Line Information (RFC5515)) as defined in `draft-lihawi-ancp-protocol-access-extension` which can be optionally enabled using this configuration attribute.<br><br>RFC and draft compliance are partial except as specified.<br><br>The value DRAFT-LIHAWI-00 enables PON attributes based on the definition in `draft-lihawi-ancp-protocol-access-extension-00` whereas DRAFT-LIHAWI-04 uses `draft-lihawi-ancp-protocol-access-extension-04`.<br><br>**Default**<br>DISABLED, Values: DRAFT-LIHAWI-00, DRAFT-LIHAWI-04 |
| `connect-speed-update` | Enable `L2TP Connect-Speed-Update-Notification` (CSUN) requests as defined in RFC5515. For more information, see Connect-Speed-Update-Notification (CSUN).<br><br>CSUN is an L2TP control message sent by the LAC to the LNS to provide updates on transmit and receive connection speeds for one or more sessions. By default, this feature is disabled but can be enabled using this configuration. Default: False. |

API To access the RESTCONF API that corresponds to this CLI, click here.

The following example shows a typical L2TPv2 LAC configuration profile. It outlines a typical L2TPv2 LAC profile, defining essential parameters such as session limits, heartbeat intervals, client identification, and security settings.

This configuration example defines an L2TPv2 LAC (L2TP Access Concentrator) profile named 'l2tp-default'. The `profile-name` is set 'l2tp-default'. This `session-limit` parameter sets the maximum number of sessions that can be established within this L2TP tunnel. The value '4000' indicates that up to 4000 sessions are allowed. The `hello-interval` is set to '60' which indicates the interval (in

seconds) for sending 'hello' messages to check the status of the L2TP tunnel.

The `client-name` is specified as 'BNG', which is the name of the client of the L2TP tunnel. The `client-ipv4` is set to '198.51.100.200' which is the 'BNG' IP address. The `hide-authentication` parameter determines whether the authentication credentials for the L2TP tunnel should be hidden. Setting it to `true` indicates the credentials are hidden.

The parameter `service-label` is set to '1234'. It assigns an MPLS service label to the L2TP tunnel. The value '1234' is used as the service label for this tunnel.

```
supervisor@switch: cfg> show config access l2tp-profile l2tp-default
{
  "rtbrick-config:l2tp-profile": {
    "profile-name": "l2tp-default",
    "session-limit": 4000,
    "hello-interval": 60,
    "client-name": "BNG",
    "client-ipv4": "198.51.100.200",
    "hide-authentication": true
    "service-label": 1234
  }
}
```

**Configuring L2TP over MPLS**

L2TP over MPLS requires a dedicated L2TP service label which must be defined manually. First, you must assign a specific MPLS label for L2TP services. This is done in the L2TP profile configuration. Next, you must configure BGP to advertise this label. This involves configuring a routing policy that specifies how to handle the L2TP service label in BGP updates.

Following is an example command for L2TP configuration with L2TP service label.

```
set access l2tp-profile l2tp-default service-label 1234
```

Advertising this label through BGP must be configured manually as shown in the following example. The exact policy configuration depends on the actual network and existing policy concept.

In the following configuration, the policy named L2TP_MPLS contains rules for traffic related to L2TP over MPLS. The policy consists of multiple rules, each specifying conditions for matching traffic and actions to take.

In the 'Ordinal 1' match rule, it specifies the `Type` as 'ipv4-prefix', `Value-Type` as 'discrete', `Match-Type` as 'exact' and `Value` is set to '198.51.100.200/24'. This rule matches traffic that has an exact IPv4 prefix of 198.51.100.200/24.

In the Action Rule, it specifies `Type` as 'label', `Operation` as 'overwrite' and `Value: label` is set to '1337,bos:1'. If the traffic matches the IPv4 prefix, the action overwrites the MPLS label with '1337' and sets the BOS bit to '1'.

In the 'Ordinal 2', it defines the action rule. After applying the label overwrite action in 'Ordinal 1', this rule permits the traffic to continue through the network with the updated label.

```
supervisor@switch: cfg> show config policy
{
    "rtbrick-config:policy": {
      "statement": [
        {
          "name": "L2TP_MPLS",
          "ordinal": [
            {
              "ordinal": 1,
              "match": {
                "rule": [
                  {
                    "rule": 1,
                    "type": "ipv4-prefix",
                    "value-type": "discrete",
                    "match-type": "exact",
                    "value": "198.51.100.200/24"
                  }
                ]
              },
              "action": {
                "rule": [
                  {
                    "rule": 1,
                    "type": "label",
                    "operation": "overwrite",
                    "value": "label:1337,bos:1"
                  }
                ]
              }
            },
            {
              "ordinal": 2,
              "action": {
                "rule": [
                  {
                    "rule": 1,
                    "operation": "return-permit"
                  }
                ]
              }
            }
          ]
        }
      ]
    }
  }
```

In the following configuration, the instance name is set as 'internet' and the Address Family is specified as 'ipv4 and unicast'. The export policy is set to L2TP_MPLS, indicating that the label defined in the L2TP_MPLS policy will be advertised for this instance.

```
supervisor@switch: cfg> show config instance internet
{
  "rtbrick-config:instance": {
    "name": "internet",
    "address-family": [
      {
        "afi": "ipv4",
        "safi": "unicast",
        "policy": {
          "export": "L2TP_MPLS"
        }
      }
    ]
  }
}
```

#### Configuring L2TP LCP Classification

RBFS supports L2TP LCP (Link Control Protocol) classification. It allows the system to identify LCP packets within an L2TP tunnel and handle them differently by tagging LCP control packets for special processing.

The feature allows for the prioritization of LCP Echo-Request and Echo-Reply messages which ensure stable L2TP sessions. The L2TP LCP Classification support only downstream traffic, that is traffic traversing from the LNS (L2TP Network Server) to the subscriber.

L2TP LCP classification is supported only on Q2C-based platforms.

**Syntax:**

**set forwarding-options acl l3v4 rule** <rule-name> **ordinal** <number> **match direction** [**ingress** | **egress**]
**set forwarding-options acl l3v4 rule** <rule-name> **ordinal** <number> **match l2tp-ppp-lcp true**
**set forwarding-options acl l3v4 rule** <rule-name> **ordinal** <number> **match destination-ipv4-prefix** <destination-ipv4-prefix>
**set forwarding-options acl l3v4 rule** <rule-name> **ordinal** <number> **action forward-class** [**class-0** | **class-1** | **class-2** | **class-3** | **class-4** | **class-5** | **class-6** | **class-7** | **class-all**]
**set forwarding-options acl l3v4 rule** <rule-name> **ordinal** <number> **priority** <value>

| Options | Description |
|---|---|
| <number> | Specifies the ordinal number. |

| Options | Description |
|---|---|
| `match direction` | Traffic direction. Only ingress is supported. |
| `l2tp-ppp-lcp [true / false]` | Specify 'true' to classify the LCP control packets. |
| `action forward-class` | Specify the forward class for action. |
| `priority` | Prioritize this ACL rule against others. |

Example:

This configuration creates an IPv4 ACL rule named 'l2tp_match_lcp' to identify ingress L2TP PPP LCP packets destined for the IPv4 prefix 131.0.0.1/32. Matching packets are assigned to forwarding class class-4 to provide differentiated traffic handling. The rule is configured with ordinal 1 and a priority of 100, which determine its position and evaluation order within the ACL.

```
set forwarding-options acl l3v4 rule l2tp_match_lcp
set forwarding-options acl l3v4 rule l2tp_match_lcp ordinal 1
set forwarding-options acl l3v4 rule l2tp_match_lcp ordinal 1 match direction ingress
set forwarding-options acl l3v4 rule l2tp_match_lcp ordinal 1 match l2tp-ppp-lcp true
set forwarding-options acl l3v4 rule l2tp_match_lcp ordinal 1 match destination-ipv4-
prefix 131.0.0.1/32
set forwarding-options acl l3v4 rule l2tp_match_lcp ordinal 1 action forward-class
class-4
set forwarding-options acl l3v4 rule l2tp_match_lcp ordinal 1 priority 100
```

```
"rtbrick-config:acl": {
    "l3v4": {
      "rule": [
        {
          "rule-name": "l2tp_match_lcp",
          "ordinal": [
            {
              "ordinal-value": 1,
              "match": {
                "destination-ipv4-prefix": "131.0.0.1/32",
                "direction": "ingress",
                "l2tp-ppp-lcp": "true"
              },
              "action": {
                "forward-class": "class-4"
              },
              "priority": 100
            }
          ]
        }
      ]
    }
  }
```

```
}
```

### L2TP Tunnel Pool Configuration

The configuration of the Layer 2 Tunnel Protocol (L2TPv2) pool is optional for subscriber management. It is necessary only if you want to enable L2TP tunneling.

The L2TP pool configuration helps to efficiently organize and manage multiple LNS servers. It allows you to define these sets locally to ensure that L2TP tunnels are directed to the appropriate servers.

**Syntax:**

**set access l2tp-pool** <pool-name> <client-name> <server-name> **client-ipv4** <client-ipv4>
**set access l2tp-pool** <pool-name> <client-name> <server-name> **preference** [0 - 65535]
**set access l2tp-pool** <pool-name> <client-name> <server-name> **secret-encrypted-text** <secret-encrypted-text>
**set access l2tp-pool** <pool-name> <client-name> <server-name> **secret-plain-text** <secret-plain-text>
**set access l2tp-pool** <pool-name> <client-name> <server-name> **server-ipv4** <server-ipv4>
**set access l2tp-pool** <pool-name> <client-name> <server-name> **session-limit** [1 - 65535]

#### Configuring the L2TP Tunnel Pool

The following command and options allow you to configure an L2TP tunnel pool.

| Attribute | Description |
|---|---|
| `<pool-name>` | Specifies the name of the L2TP pool, which groups L2TP client-server mappings for tunnel establishment. |
| `<client-name>` | Specifies the local L2TP tunnel client (LAC) hostname, representing the client initiating L2TP tunnels. |
| `<server-name>` | Specifies the remote L2TP tunnel server (LNS) hostname, which terminates L2TP tunnels and provides subscriber services. |
| `client-ipv4` | Specifies the IPv4 address of the local L2TP tunnel client (LAC) IPv4 address used for establishing the tunnel. |
| `server-ipv4` | Specifies the remote L2TP tunnel server (LNS) IPv4 address to which tunnels are established. |
| `preference` | Defines the priority of the LNS within the pool. Lower values indicate higher preference when selecting a server. Default: 0, Range: 1 - 65535. |
| `session-limit` | Tunnels with a session limit reached are not considered for further sessions. This limit has precedence over the default session limit specified in the `l2tp-profile`. Default: 64000, Range: 1 - 65535. |
| `secret-encrypted-text` | Configures the shared secret in encrypted form for secure authentication between the LAC and LNS. |

| Attribute | Description |
| --- | --- |
| `secret-plain-text` | Configures the shared secret in plain text format. L2TP tunnel secret can be provided as plaintext or an already encrypted text. |

The following example shows a local pool with two LNS severs.

This configuration sets an L2TP pool named 'lns-pool-example' with two LNS servers (LNS1 and LNS2). It defines how the device will manage L2TP tunnels. In this example, there are two different LNS server entries for two different LNS servers, LNS1 and LNS2. Both entries belong to the same pool named 'lns-pool-example'. Each entry in the `l2tp-pool` array defines the configuration for an individual LNS server.

The `client-name` is specified as BNG, which indicates that the client will initiate L2TP tunnels to the listed LNS servers. The `server-name` specifies the name of the LNS. Each entry has a unique server name, LNS1 for the first entry and LNS2 for the second.

The `client-ipv4` specifies the IPv4 address of the L2TP client, that is BNG. In both entries, the LAC's IP address is '198.51.100.200'. The `server-ipv4` specifies the IPv4 address of the LNS. The first entry has '198.51.100.219' for LNS1, and the second entry has '198.51.100.220' for LNS2.

The `secret-encrypted-text` parameter specifies the shared secret used for authenticating the L2TP tunnel between the LAC and LNS. The value is encrypted, and the secret is the same in both entries: '$21e4946e31b406de98b3077aef03ed5a7'.

The `preference` parameter sets the preference value for the LNS server. A lower value generally indicates a higher preference, but in this case, both entries have the same preference value of 1000. The `session-limit` parameter specifies the maximum number of sessions (L2TP tunnels) that can be established with the LNS server. Both entries have a session limit of 1000.

```
supervisor@switch: cfg> show config access
{
  "rtbrick-config:access": {
    "l2tp-pool": [
      {
        "pool-name": "lns-pool-example",
        "client-name": "BNG",
        "server-name": "LNS1",
        "client-ipv4": "198.51.100.200",
        "server-ipv4": "198.51.100.219",
        "secret-encrypted-text": "$21e4946e31b406de98b3077aef03ed5a7",
        "preference": 1000,
        "session-limit": 1000
      },
      {
        "pool-name": "lns-pool-example",
        "client-name": "BNG",
        "server-name": "LNS2",
        "client-ipv4": "198.51.100.200",
        "server-ipv4": "198.51.100.220",
```

```
        "secret-encrypted-text": "$21e4946e31b406de98b3077aef03ed5a7",
        "preference": 1000,
        "session-limit": 1000
      }
    ]
  }
}
```

To access the RESTCONF API that corresponds to this CLI, click here.

### User Profile Configuration

This section describes how to configure locally defined user profiles for subscriber authentication. While external servers such as RADIUS are typically used, the system also supports local authentication using these profiles. User profiles allow you to define credentials and session parameters directly on the device. This provides an alternative authentication method, especially useful in small deployments or testing scenarios.

#### Configuring User Profile

**Syntax:**

**set access user-profile** <user-name> **l2tp-pool-name** <l2tp-pool-name>
**set access user-profile** <user-name> **password-encrypted-text** <password-encrypted-text>
**set access user-profile** <user-name> **password-plain-text** <password-plain-text>
**set access user-profile** <user-name> **tunnel-type [PPPoE | L2TP]**

The following command and options are used to configure a user profile. A user profile also includes authentication details.

| Attribute | Description |
|---|---|
| `-` | Without any option, the command sets a user profile. |
| `user-name` | Specifies the username for which the user profile is configured. |
| `l2tp-pool-name` | Associates an L2TP pool used for assigning tunnel endpoints to the user. |
| `password-encrypted-text` | Specifies the user password in encrypted form for secure authentication. |
| `password-plain-text` | Specifies the user password in plain text, which is internally encrypted before storage. |
| `tunnel-type` | Specifies the tunnel type used for the subscriber session (for example, L2TP). Default: PPPoE. Values: PPPoE or L2TP. |

The following configuration shows the user name specified as 'user@rtbrick.com'. The `password-encrypted-text` attribute enables you to set the user's password in an encrypted format. The `tunnel-type` is set to PPPoE.

```
supervisor@switch: cfg> show config access user-profile user@rtbrick.com
{
  "rtbrick-config:user-profile": {
    "user-name": "user@rtbrick.com",
    "password-encrypted-text": "$243a1341f44f54888cdd385b9f40513f1",
    "tunnel-type": "PPPoE"
  }
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

### IP Address Pool Configuration

IP address pools are crucial for assigning IP addresses to subscribers, and configuring these pools is an essential for subscriber management.

The following image illustrates the IP address pool configuration and how they are associated with the other subscriber management tasks.

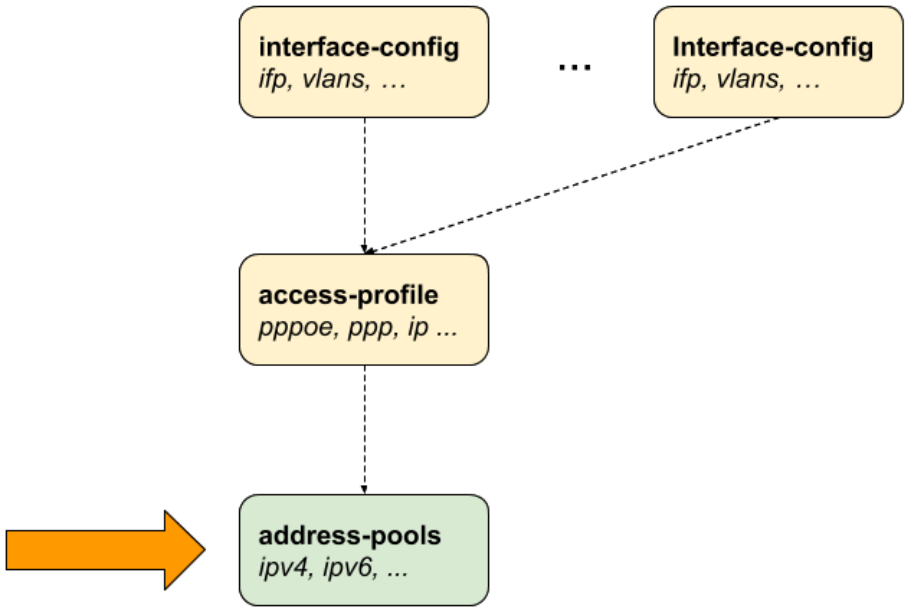



*Figure 73. IP Address Pool Configuration*

#### RBFS IP Address Pool

RBFS allows you to manage address pools with more flexibility. It enables you to modify or remove pools without disrupting active subscriber sessions. If an address pool is deleted, subscribers who have already been assigned IP addresses from that pool can continue using those addresses until their session ends. The system does not immediately revoke the IP addresses assigned to active sessions.

Imagine a subscriber has been allocated an IP address from a specific address pool. If that pool is later deleted, the subscriber's connection remains unaffected until their session ends.

It displays an error message when a pool that is part of a pool chain is deleted, or when a non-

existent pool is attached to an access profile. It ensures configuration consistency when 'next-pool-name' is not defined or invalid.

The following is an example of a generated error message.

```
Jan 16 04:28:37: Commit failed. Edit and try again or discard changes: application
operation-failed IPv6 pool referenced in access-profile does not exist
```

Ensure that the deleted address pool is not reallocated elsewhere until all assigned addresses are no longer in use. This prevents IP address conflicts.

#### Duplicate Address Detection

RBFS provides an optional duplicate address detection feature that prevents a new session from using an IP address already in use by another active session. This is to avoid conflicts. However, when an address pool is moved to a different BNG, the duplicate address detection feature may not work as expected.

Migrating address pools across different BNGs could lead to address conflicts, if not managed carefully.

#### Configuring Address Pool

The following command and options allows you to configure both IPv4 and IPv6 address pools for subscriber access.

**set access pool** <pool-name> **ipv4-address**
**set access pool** <pool-name> **ipv6-prefix**
**set access pool** <pool-name> **next-pool-name** <next-pool-name>
**set access pool** <pool-name> **gateway-ifl** <gateway-ifl>

| Attribute | Description |
|---|---|
| `<pool-name>` | Specifies the unique name used to identify the address pool. |
| `ipv4-address` | Configures the IPv4 address range and related parameters for assigning IPv4 addresses to subscribers. |
| `ipv6-prefix` | Configures the IPv6 prefix pool used for delegating IPv6 prefixes to subscribers. |
| `next-pool-name` | Specifies an alternate address pool to be used when the current pool is exhausted, enabling hierarchical or fallback allocation. |

| Attribute | Description |
| --- | --- |
| gateway-ifl <gateway-ifl> | Specifies the logical interface (IFL) gateway to associate with the IPv4 pool. This enables the subscriber daemon to dynamically select the correct default gateway for chained local IPv4 pools based on the allocated IP pool range.<br>It is supported only for IPv4 address pools. |

The following example shows typical IPv4 address and IPv6 prefix pools. It shows three pools: one for IPv4 addresses and two for IPv6 prefixes. Each pool is defined by a name, and a range of addresses or prefixes is specified using the 'low' and 'high' values. These pools are used for dynamically assigning IP addresses or prefixes to devices or subscribers, depending on the protocol (IPv4 or IPv6).

The IPv4 address pool, named 'ipv4-local', includes the address range from ‘198.51.100.76’ to ‘198.51.100.117’. This pool defines a range of IPv4 addresses from 198.51.100.76 to 198.51.100.117. These addresses can be dynamically assigned to devices or subscribers that require IPv4 connectivity.

The specified second pool is ‘ipv6-local’ that defines a range of IPv6 prefixes, from ‘2001:db8:0:79::/32’ to ‘2001:db8:0:139::/32’. These prefixes can be assigned to devices or subscribers that require IPv6 connectivity.

The IPv6 Prefix Delegation Pool is specified as ‘ipv6pd-local’. It defines the range of IPv6 from 2001:db8:0:1::/32 to ‘2001:db8:0:100::/32’. This pool is similar to the previous IPv6 pool, but is specifically meant for IPv6 Prefix Delegation.

```
supervisor@switch: cfg> show config access
{
  "rtbrick-config:access": {
    "pool": [
      {
        "pool-name": "ipv4-local",
        "ipv4-address": {
          "low": "198.51.100.76",
          "high": "198.51.100.117"
        }
      },
      {
        "pool-name": "ipv6-local",
        "ipv6-prefix": {
          "low": "2001:db8:0:79::/32",
          "high": "2001:db8:0:139::/32"
        }
      },
      {
        "pool-name": "ipv6pd-local",
        "ipv6-prefix": {
          "low": "2001:db8:0:1::/32",
```

```
          "high": "2001:db8:0:100::/32"
        }
      }
    ],
  }
}
```

The following configuration creates two chained access pools with different gateway IFLs and IPv4 address ranges for IPoE subscribers.

```
set interface lo-0/0/0 unit 1 address ipv4 192.0.2.1/32
set interface lo-0/0/0 unit 2 address ipv4 198.51.100.1/32
set access pool p1
set access pool p1 next-pool-name p2
set access pool p1 gateway-ifl lo-0/0/0/1
set access pool p1 ipv4-address low 192.0.2.2
set access pool p1 ipv4-address high 192.0.2.254
set access pool p1 ipv4-address subnet-mask 255.255.255.0
set access pool p2
set access pool p2 gateway-ifl lo-0/0/0/2
set access pool p2 ipv4-address low 198.51.100.2
set access pool p2 ipv4-address high 198.51.100.254
set access pool p2 ipv4-address subnet-mask 255.255.255.0
```

```
{
  "ietf-restconf:data": {
    "rtbrick-config:interface": [
      {
        "name": "lo-0/0/0",
        "unit": [
          {
            "unit-id": 1,
            "address": {
              "ipv4": [
                {
                  "prefix4": "192.0.2.1/32"
                }
              ]
            }
          },
          {
            "unit-id": 2,
            "address": {
              "ipv4": [
                {
                  "prefix4": "198.51.100.1/32"
                }
              ]
```

```
            }
          }
        ]
      }
    ],
    "rtbrick-config:access": {
      "pool": [
        {
          "pool-name": "p1",
          "next-pool-name": "p2",
          "gateway-ifl": "lo-0/0/0/1",
          "ipv4-address": {
            "low": "192.0.2.2",
            "high": "192.0.2.254",
            "subnet-mask": "255.255.255.0"
          }
        },
        {
          "pool-name": "p2",
          "gateway-ifl": "lo-0/0/0/2",
          "ipv4-address": {
            "low": "198.51.100.2",
            "high": "198.51.100.254",
            "subnet-mask": "255.255.255.0"
          }
        }
      ]
    }
  }
}
```


To access the RESTCONF API that corresponds to this CLI, click here.

#### Configuring IPv4 Address Pools

The following command and options allow you to configure an IPv4 address pool.

**Syntax:**

**set access pool** <pool-name> **ipv4-address high** <high>
**set access pool** <pool-name> **ipv4-address low** <low>
**set access pool** <pool-name> **ipv4-address subnet-mask** <subnet-mask>

| Attribute | Description |
|---|---|
| high | Specifies the highest IPv4 address in the local address pool range that can be assigned to subscribers. |
| low | Specifies the lowest IPv4 address in the local address pool range that can be assigned to subscribers. |

| Attribute | Description |
|---|---|
| subnet-mask | Defines the subnet mask for the IPv4 address pool, determining the size and boundary of the address range. |

```
supervisor@rtbrick.net: cfg> show config access pool ipoe_v4_pool1
{
  "rtbrick-config:pool": [
    {
      "pool-name": "ipoe_v4_pool1",
      "next-pool-name": "ipoe_v4_pool2",
      "ipv4-address": {
        "low": "11.100.128.2",
        "high": "11.100.128.6"
      }
    }
  ]
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Configuring IPv6 Prefix Pools

The following command and options allow you to configure an IPv6 prefix pools.

**Syntax:**

**set access pool** <pool-name> **ipv6-prefix high** <high>
**set access pool** <pool-name> **ipv6-prefix low** <low>
**set access pool** <pool-name> **next-pool-name** <next-pool-name>

| Attribute | Description |
|---|---|
| high | Specifies the highest IPv4 address in the local address pool range that can be assigned to subscribers. |
| low | Specifies the lowest IPv6 address in the local address pool range that can be assigned to subscribers. |

IPv6 prefixes must be at least /64 or larger (/56, /48, ...) or /128.

```
supervisor@rtbrick.net: cfg> show config access pool ipoe_v6_pool3
{
  "rtbrick-config:pool": [
    {
      "pool-name": "ipoe_v6_pool3",
      "ipv6-prefix": {
        "low": "fc65:1:1:b::/64",
```

```
        "high": "fc65:1:1:32::/64"
      }
    }
  ]
}
```

#### Configuring Linked Pools (Pool Chain)

RBFS enables pool linking. You can link or chain multiple IP address pools so that when a pool gets exhausted, the next pool defined will start performing address allocation. Pool linking allows the creation of discontinuous address pools connected together in a sequence, creating an extended pool. The pools are connected using the `next-pool-name` attribute.

**Syntax:**

**set access pool** <pool-name> **next-pool-name** <next-pool-name>

Example:

`A -> B -> C -> D`

```
set access pool pool-A next-pool pool-B
set access pool pool-B next-pool pool-C
set access pool pool-C next-pool pool-D
```

The actual address pool assigned to a subscriber is determined by the access configuration profile or RADIUS, starting with the specified pool in the chain. If that pool is full, the system requests the next pool in sequence, continuing until an available pool is found or the end of the chain is reached. To prevent looping, RBFS halts automatically if it detects that any pool in the chain has been accessed twice.

Additionally, the system can continuously cycle through all available pools in the chain, even if allocation starts from a pool in the middle. This ensures every pool in the sequence gets considered till finding an available address.

The following configuration commands link a set of IP address pools in a cycle order. The 'pool-A' is linked with the 'pool-B', 'pool-C' is set as a next pool after the 'pool-B'. 'Pool-D' is specified as next pool for 'pool-C'. 'Pool-A' is set as the next pool for 'pool-A'.

`A -> B -> C -> D -> A`

```
set access pool pool-A next-pool pool-B
set access pool pool-B next-pool pool-C
set access pool pool-C next-pool pool-D
set access pool pool-D next-pool pool-A
```

```
supervisor@rtbrick.net: cfg> show config access pool ipoe_v4_pool1 next-pool-name
{
  "rtbrick-config:next-pool-name": "ipoe_v4_pool2"
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

**Control Plane Outbound Marking Configuration**

Control Plane Outbound Marking refers to the process of marking control plane packets as they exit the network device. This marking ensures that packets are processed in accordance with defined network policies, facilitating consistent and reliable traffic handling.

To mark all outbound control plane packets, configure the appropriate settings under `forwarding-options`. The following example demonstrates this configuration.

With these settings, control plane traffic is reliably identified and handled according to specified criteria, supporting an optimized network environment.

The following configuration marks outbound control plane packets with specific attributes, such as priority or type of service, to simplify their identification within the network.

This configuration defines how outbound control plane packets are marked by protocol under `forwarding-options`. Control Plane QoS settings ensure that control traffic is marked and prioritized appropriately. The `outbound-marking` specifies marking criteria for control plane traffic. In the example, the `protocol` and `remark-types` options are configured for L2TPv2, PPP, and RADIUS protocols.

Each protocol uses distinct marking settings:

- L2TPv2 packets are marked with a ToS codepoint of 64 to ensure they are handled with the designated priority as they traverse the network.
- PPP packets are marked with a p-bit (priority bit) value of 6, indicating their priority within the network.
- Similar to L2TPv2, RADIUS packets are marked with a ToS codepoint of 64, ensuring consistent priority treatment across protocols.

```
supervisor@switch: cfg> show config forwarding-options
{
    "rtbrick-config:forwarding-options": {
      "class-of-service": {
        "control-plane-qos": {
          "outbound-marking": {
            "protocol": [
              {
                "protocol": "l2tpv2",
                "remark-type": "tos",
                "codepoint": 64
```

```
            },
            {
              "protocol": "ppp",
              "remark-type": "p-bit",
              "codepoint": 6
            },
            {
              "protocol": "radius",
              "remark-type": "tos",
              "codepoint": 64
            }
          ]
        }
      }
    }
  }
}
```

### Configuration Examples

#### PPPoE

The following example shows a PPPoE configuration for VLAN mode 1:1 with IPv4 and IPv6 enabled, authenticated via RADIUS.

```
{
  "ietf-restconf:data": {
    "rtbrick-config:access": {
      "aaa-profile": [
        {
          "profile-name": "aaa-radius",
          "session-timeout": 0,
          "idle-timeout": 0,
          "aaa-radius-profile": "radius-default",
          "authentication": {
            "order": "RADIUS"
          },
          "accounting": {
            "order": "RADIUS",
          }
        }
      ],
      "radius-profile": [
        {
          "profile-name": "radius-default",
          "nas-identifier": "BNG",
          "nas-port-type": "Ethernet",
          "authentication": {
            "radius-server-profile-name": [
              "radius-server-1",
```

```
            "radius-server-2"
            ]
          },
          "accounting": {
            "radius-server-profile-name": [
              "radius-server-1",
              "radius-server-2"
              ],
            "stop-on-reject": "true",
            "stop-on-failure": "true",
            "accounting-on-off": "true",
            "accounting-on-wait": "true",
            "accounting-backup": "true",
            "accounting-backup-max": 86400
          }
        }
      ],
      "radius-server": [
        {
          "server-name": "radius-server-1",
          "address": "198.51.100.101",
          "source-address": "198.51.100.200",
          "secret-encrypted-text": "$21e4946e31b406de98b3077aef03ed5a7",
          "authentication": {
            "enable": "true"
          },
          "accounting": {
            "enable": "true"
          },
          "coa": {
            "enable": "true"
          }
        },
        {
          "server-name": "radius-server-2",
          "address": "198.51.100.102",
          "source-address": "198.51.100.200",
          "secret-encrypted-text": "$21e4946e31b406de98b3077aef03ed5a7",
          "authentication": {
            "enable": "true"
          },
          "accounting": {
            "enable": "true"
          },
          "coa": {
            "enable": "true"
          }
        }
      ],
      "access-profile": [
        {
```

```
        "profile-name": "pppoe-dual",
        "protocol": {
          "pppoe": {
            "enable": "true",
            "session-protection": {
              "enable": "true"
            },
            "vlan-priority": 6
          },
          "ppp": {
            "lcp": {
              "authentication-protocol": "PAP_CHAP",
              "echo-interval": 30,
              "echo-max-retransmit": 3,
              "echo-enable": "true"
            },
            "ipcp": {
              "enable": "true",
              "source-ifl": "lo-0/0/0/1"
            },
            "ip6cp": {
              "enable": "true"
            }
          },
          "ra": {
            "enable": "true",
            "interval": 60
          },
          "dhcpv6": {
            "enable": "true"
          },
          "l2tp": {
            "tunnel-profile": "l2tp-default"
          }
        },
        "address-family": {
          "ipv4": {
            "enable": "true",
            "primary-dns": "198.51.100.103",
            "secondary-dns": "198.51.100.104",
            "instance": "default"
          },
          "ipv6": {
            "enable": "true",
            "primary-dns": "2001:db8:0:100::",
            "secondary-dns": "2001:db8:0:104::",
            "instance": "default"
          }
        }
      }
    ],
```

```
      "interface": {
        "double-tagged": [
          {
            "interface-name": "ifl-0/0/1",
            "outer-vlan-min": 1,
            "outer-vlan-max": 4094,
            "inner-vlan-min": 7,
            "inner-vlan-max": 7,
            "access-type": "PPPoE",
            "access-profile-name": "pppoe-dual",
            "aaa-profile-name": "aaa-radius"
          }
        ]
      },
      "l2tp-profile": [
        {
          "profile-name": "l2tp-default",
          "session-limit": 4000,
          "client-name": "BNG",
          "client-ipv4": "198.51.100.200",
          "hide-authentication": true
        }
      ]
    },
    "rtbrick-config:interface": [
      {
        "name": "ifl-0/0/1",
        "description": "Access",
        "host-if": "eth0"
      },
      {
        "name": "ifl-0/0/2",
        "description": "Core",
        "host-if": "eth1",
        "unit": [
          {
            "unit-id": 1,
            "address": {
              "ipv4": [
                {
                  "prefix4": "198.51.100.33/24"
                }
              ],
              "ipv6": [
                {
                  "prefix6": "2001:db8:0:32::/32"
                }
              ]
            }
          }
        ]
```

```
      },
      {
        "name": "lo-0/0/0",
        "unit": [
          {
            "unit-id": 1,
            "address": {
              "ipv4": [
                {
                  "prefix4": "198.51.100.200/24"
                }
              ]
            }
          }
        ]
      }
    ]
  }
}
```

**IPoE**

The following example shows an IPoE configuration for VLAN mode 1:1 with IPv4 and IPv6 enabled, authenticated via RADIUS.

```
{
  "ietf-restconf:data": {
    "rtbrick-config:access": {
      "aaa-profile": [
        {
          "profile-name": "aaa-radius",
          "session-timeout": 0,
          "idle-timeout": 0,
          "aaa-radius-profile": "radius-default",
          "authentication": {
            "order": "RADIUS"
          },
          "accounting": {
            "order": "RADIUS",
          }
        }
      ],
      "radius-profile": [
        {
          "profile-name": "radius-default",
          "nas-identifier": "BNG",
          "nas-port-type": "Ethernet",
          "authentication": {
            "radius-server-profile-name": [
              "radius-server-1",
```

```
            "radius-server-2"
            ]
          },
          "accounting": {
            "radius-server-profile-name": [
              "radius-server-1",
              "radius-server-2"
              ],
            "stop-on-reject": "true",
            "stop-on-failure": "true",
            "accounting-on-off": "true",
            "accounting-on-wait": "true",
            "accounting-backup": "true",
            "accounting-backup-max": 86400
          }
        }
      ],
      "radius-server": [
        {
          "server-name": "radius-server-1",
          "address": "198.51.100.101",
          "source-address": "198.51.100.200",
          "secret-encrypted-text": "$21e4946e31b406de98b3077aef03ed5a7",
          "authentication": {
            "enable": "true"
          },
          "accounting": {
            "enable": "true"
          },
          "coa": {
            "enable": "true"
          }
        },
        {
          "server-name": "radius-server-2",
          "address": "198.51.100.102",
          "source-address": "198.51.100.200",
          "secret-encrypted-text": "$21e4946e31b406de98b3077aef03ed5a7",
          "authentication": {
            "enable": "true"
          },
          "accounting": {
            "enable": "true"
          },
          "coa": {
            "enable": "true"
          }
        }
      ],
      "access-profile": [
        {
```

```
          "profile-name": "ipoe-dual",
          "protocol": {
            "dhcp": {
              "enable": "true",
              "mode": "server"
            },
            "dhcpv6": {
              "enable": "true",
              "mode": "server"
            },
          },
          "address-family": {
            "ipv4": {
              "enable": "true",
              "pool-name":"ipoe",
              "primary-dns": "198.51.100.103",
              "secondary-dns": "198.51.100.104",
              "instance": "default"
            },
            "ipv6": {
              "enable": "true",
              "pool-name":"ipoe-ia-na",
              "prefix-delegation-pool-name":"ipoe-ia-pd",
              "primary-dns": "2001:db8:0:100::",
              "secondary-dns": "2001:db8:0:104::",
              "instance": "default"
            }
          }
        }
      ],
      "interface": {
        "double-tagged": [
          {
            "interface-name": "ifl-0/0/1",
            "outer-vlan-min": 1,
            "outer-vlan-max": 4094,
            "inner-vlan-min": 7,
            "inner-vlan-max": 7,
            "access-type": "IPoE",
            "access-profile-name": "ipoe-dual",
            "aaa-profile-name": "aaa-radius",
            "gateway-ifl": "lo-0/0/0/1"
          }
        ]
      },
      "pool": [
        {
          "pool-name": "ipoe",
          "ipv4-address": {
            "low": "10.0.0.1",
            "high": "10.0.255.255"
```

```
          }
        },
        {
          "pool-name": "ipoe-ia-na",
          "ipv6-prefix": {
            "low": "fc66::1/128",
            "high": "fc66::ffff/128"
          }
        },
        {
          "pool-name": "ipoe-ia-pd",
          "ipv6-prefix": {
            "low": "fc66:0:100::/56",
            "high": "fc66:0:1ff:ff00::/56"
          }
        }
      ],
    },
    "rtbrick-config:interface": [
      {
        "name": "ifl-0/0/1",
        "description": "Access",
        "host-if": "eth0"
      },
      {
        "name": "ifl-0/0/2",
        "description": "Core",
        "host-if": "eth1",
        "unit": [
          {
            "unit-id": 1,
            "address": {
              "ipv4": [
                {
                  "prefix4": "198.51.100.33/24"
                }
              ],
              "ipv6": [
                {
                  "prefix6": "2001:db8:0:6423::/32"
                }
              ]
            }
          }
        ]
      },
      {
        "name": "lo-0/0/0",
        "unit": [
          {
            "unit-id": 1,
```

```
            "address": {
              "ipv4": [
                {
                  "prefix4": "198.51.100.200/24"
                }
              ]
            }
          }
        ]
      }
    ]
  }
}
```

### 5.1.3. Subscriber Management Operational Commands

Subscriber Management operational commands allow you to monitor and verify subscriber sessions established through access protocols such as PPPoE, IPoE, and L2TP. These commands provide information about subscriber states, subscriber session, interface bindings, authentication status and so on.

The following diagram provides a hierarchical view of the subscriber management operational commands used to monitor subscriber sessions across all access types (PPPoE, IPoE, and L2TP).

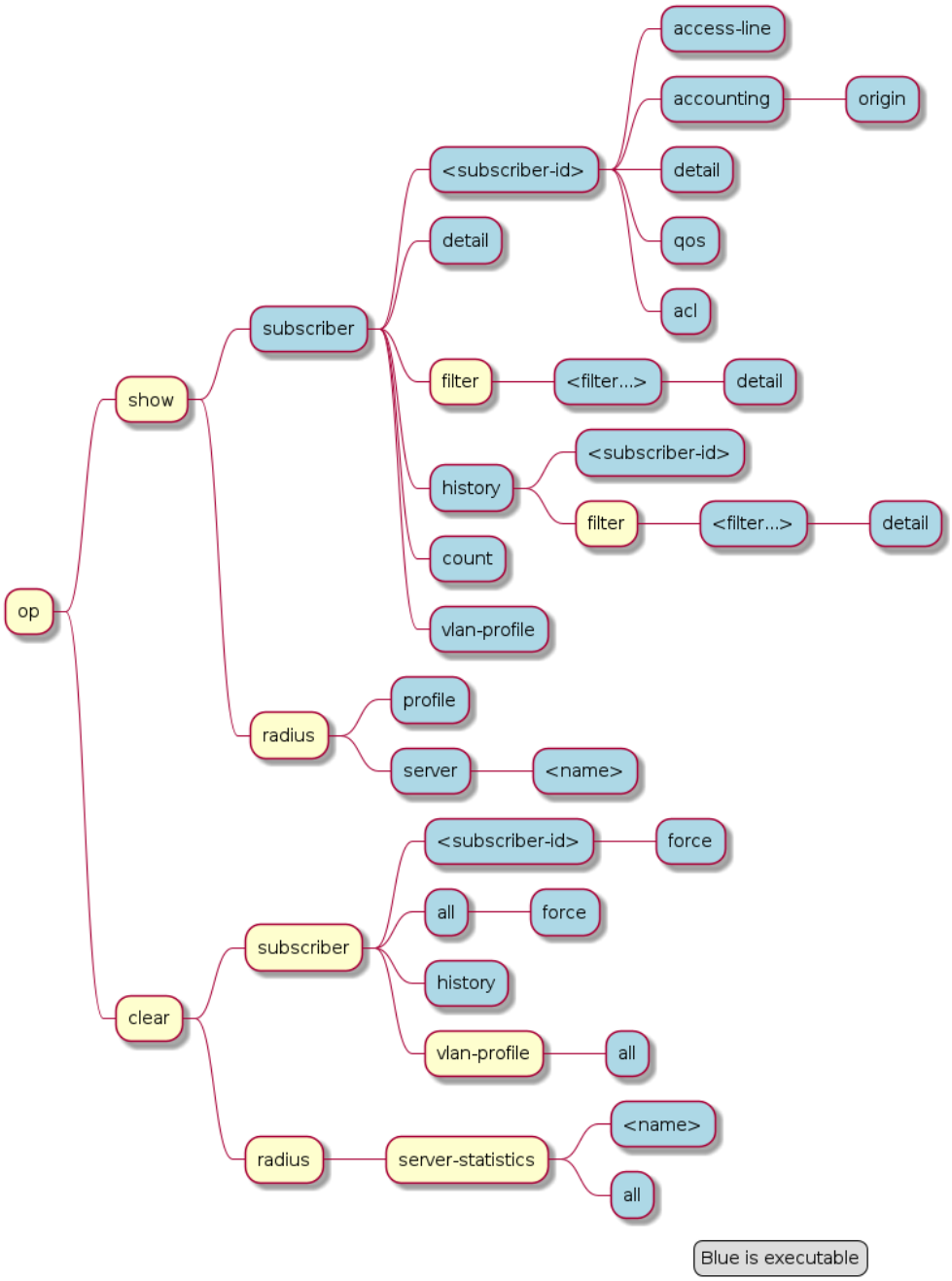



*Figure 74. Subscriber Management Operational Commands*

### Subscriber Show Commands

The show commands provide information about subscribers that include subscriber states, counts, subscriber history, subscriber pools, subscriber redundancy detailed session attributes, and termination codes. You can use various filters to narrow down to the specific details.

**Syntax:**

```
show subscriber
show subscriber count
show subscriber detail
show subscriber filter <option>
show subscriber history
```

**show subscriber pool**
**show subscriber redundancy**
**show subscriber vlan-profile**

The following table provides command options and descriptions.

| Attribute | Description |
|---|---|
| `<cr>` | Displays a summary of all active subscribers. |
| `count` | Displays aggregated total number of subscribers. |
| `detail` | Displays detailed information for all subscribers. |
| `filter` | Displays subscribers that match specified filter criteria. |
| `history` | Displays information about terminated subscribers, session details and termination reasons. |
| `pool` | Displays local address pool information, including allocation status. |
| `redundancy` | Displays subscriber redundancy details. |
| `<subscriber-id>` | Displays information for a specific subscriber. |
| `vlan-profile` | Displays configured VLAN profiles associated with subscribers. |

The following example displays the subscribers and their details.

```
supervisor@rtbrick: op> show subscriber
Subscriber-Id           Interface         VLAN       Type    State
72339069014638594       ifp-0/1/5         201:201    PPPoE   ESTABLISHED
72339069014638595       ifp-0/1/5         201:202    PPPoE   ESTABLISHED
72339069014638596       ifp-0/1/5         201:203    PPPoE   ESTABLISHED
72339069014638597       ifp-0/1/5         201:204    PPPoE   ESTABLISHED
72339069014638598       ifp-0/1/5         201:205    PPPoE   ESTABLISHED
```

#### Subscriber Details for a Specific Subscriber

These show commands provide information about status of subscribers across various access technologies. You can view the information about state transitions, from initial session setup through active and termination phases. The commands allow you to verify the status of the subscriber sessions. The state ESTABLISHED means that the session is fully operational.

The following commands are applicable to all types of subscribers that include PPPoE, L2TP, and IPoE.

**Syntax:**

**show subscriber** <subscriber-id>
**show subscriber** <subscriber-id> **access-line**
**show subscriber** <subscriber-id> **accounting**

```
show subscriber <subscriber-id> accounting origin
show subscriber <subscriber-id> acl
show subscriber <subscriber-id> acl <acl_name>
show subscriber <subscriber-id> acl <acl_name> type [l3v4 | l3v6]
show subscriber <subscriber-id> detail
show subscriber <subscriber-id> igmp
show subscriber <subscriber-id> qos
```

The following table provides command options and descriptions.

| Attribute | Description |
|---|---|
| `<cr>` | Displays a summary of the specified subscriber, including session and state information. |
| `access-line` | Displays access line details such as interface, VLAN, and physical connectivity information associated with the subscriber. |
| `accounting` | Displays accounting-related information, including usage counters and session statistics for the subscriber. |
| `acl` | Displays the details related to ACL filters applied to the subscriber. |
| `detail` | Displays detailed subscriber information, including session parameters and state. |
| `qos` | Displays QoS configuration and statistics applied to the subscriber. |

The various subscriber states are explained in the table and diagram below.

| State | Description |
|---|---|
| `INIT` | Initial subscriber state. |
| `AUTHENTICATING` | Authenticate the subscriber using the configured method. |
| `ADDRESS ALLOCATION` | Allocate (RADIUS or pool) and validate (DAD) addresses. |
| `TUNNEL SETUP` | Setup tunnel resources (L2TP or L2X). |
| `IFL SETUP` | Create subscriber IFL with corresponding QoS resources. |
| `FULL` | Wait for subscriber to be in forwarding state. Inform underlying protocols (PPPoED or IPoED) to continue with session setup. |
| `ACCOUNTING` | Start subscriber accounting and wait for response. |
| `ESTABLISHED` | The subscriber becomes ESTABLISHED after response to RADIUS Accounting-Request-Start if RADIUS accounting is enabled otherwise immediately after FULL. |
| `TERMINATING` | The subscriber remains in this state until all resources are freed and accounting stopped. This means that subscriber remain in this state until response to RADIUS Accounting-Request-Stop if RADIUS accounting is enabled. |

The following diagram illustrates the subscriber session life cycle. It shows the sequence of states

from initialization through authentication, resource allocation, and finally to an established or terminating state.

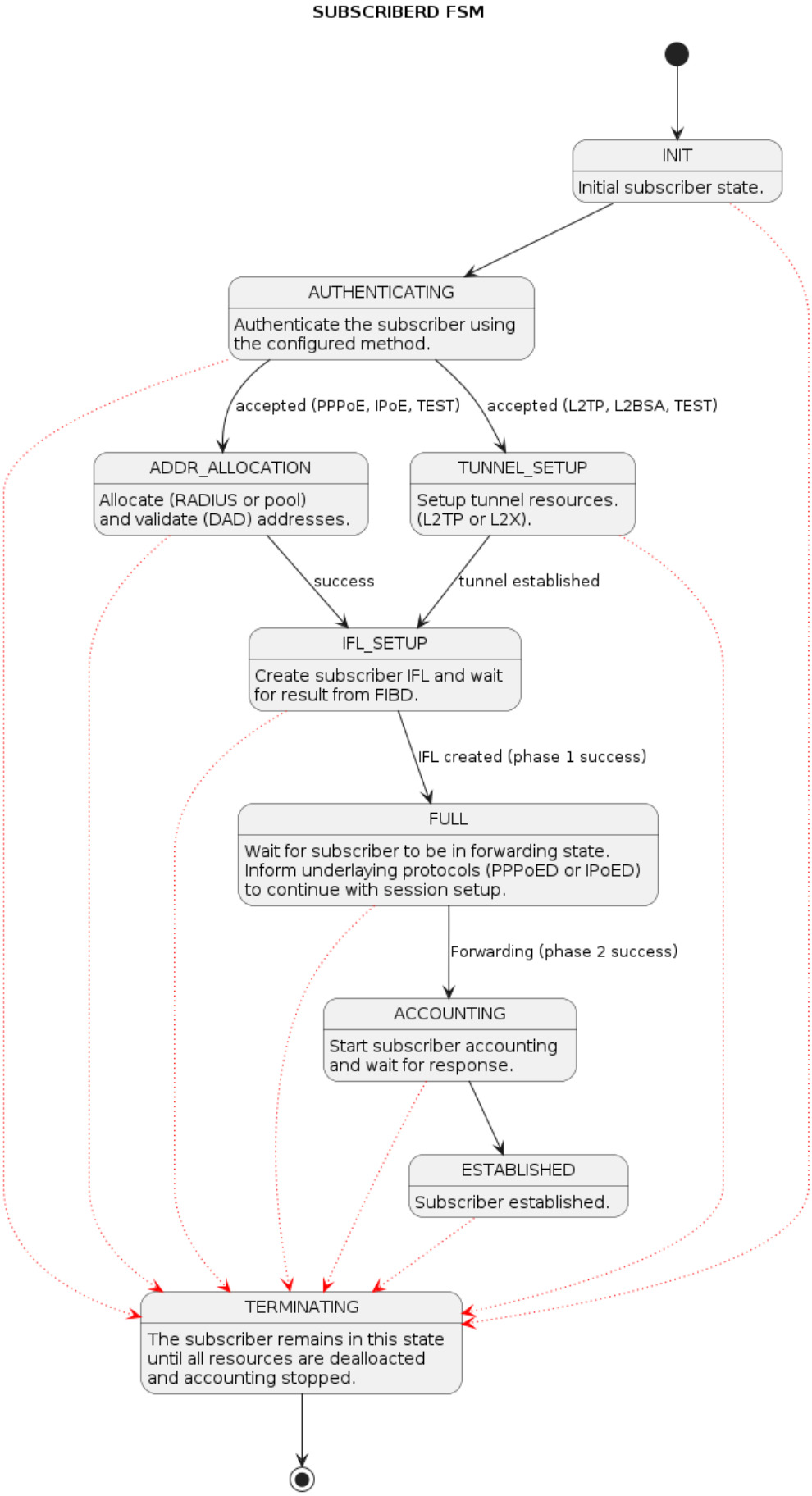



*Figure 75. Subscriber States*

Example for the list of subscribers and their details.

```
supervisor@rtbrick: op> show subscriber
Subscriber-Id           Interface         VLAN       Type    State
72339069014638594       ifp-0/1/5         201:201    PPPoE   ESTABLISHED
72339069014638595       ifp-0/1/5         201:202    PPPoE   ESTABLISHED
72339069014638596       ifp-0/1/5         201:203    PPPoE   ESTABLISHED
72339069014638597       ifp-0/1/5         201:204    PPPoE   ESTABLISHED
72339069014638598       ifp-0/1/5         201:205    PPPoE   ESTABLISHED
```

Example for the specified subscriber details.

```
supervisor@rtbrick: op> show subscriber 72339069014638594
Subscriber-Id: 72339069014638594
    Type: PPPoE
    State: ESTABLISHED
    Created: Tue Jul 14 18:08:22 GMT +0000 2026
    Interface: ifp-0/1/5
    Outer VLAN: 201 (802.1q)
    Inner VLAN: 201
    Client MAC: 02:00:00:00:00:01
    Server MAC: 5c:07:58:79:0f:06
    IFL: ppp-0/1/5/72339069014638594
    Username: 1@pppoe
    Agent-Remote-Id: DEU.RTBRICK.1
    Agent-Circuit-Id: 0.0.0.0/0.0.0.0 eth 0:1
    Access-Profile: PPPoE_ACCESS_PROFILE
    AAA-Profile: PPPoE_AAA
```

#### Subscriber Count Show Commands

You can view the total number of of PPPoE, L2TP, IPoE, and L2BSA subscribers in the different states such as 'Setup', 'Established', and 'Terminating' using the subscriber count command. This command provides information per interface and a total summary based on subscriber type.

**Syntax:**

**show subscriber count brief**
**show subscriber count interface** <interface>

The following table provides command options and descriptions.

| Attribute | Description |
|---|---|
| `count brief` | Displays a summary count of subscribers. |
| `count interface <interface>` | Displays subscriber count for the specified interface. |

Examples:

The following example for subscriber count output displays the total number of subscribers on the system and categorizes them by state (Setup, Established, Terminating, and Standby) and by access type such as PPPoE, L2TP, and IPoE.

```
supervisor@rtbrick: op> show subscriber count
                       Total        Setup        Established  Terminating  Standby
Summary                5            0            5            0            0
  PPPoE                5            0            5            0            0
  L2TP                 0            0            0            0            0
  IPoE                 0            0            0            0            0
  L2BSA                0            0            0            0            0
  Test                 0            0            0            0            0
ifp-0/1/5              5            0            5            0            0
  PPPoE                5            0            5            0            0
  L2TP                 0            0            0            0            0
  IPoE                 0            0            0            0            0
  L2BSA                0            0            0            0            0
  Test                 0            0            0            0            0
```

The command `show subscriber count brief` provides a filter to the existing `subscriber count` command, excluding lines with zero counts.

```
supervisor@rtbrick: op> show subscriber count brief
                       Total        Setup        Established  Terminating  Standby
Summary                5            0            5            0            0
  PPPoE                5            0            5            0            0
ifp-0/1/5              5            0            5            0            0
  PPPoE                5            0            5            0            0
```

Example for subscriber count for specified interface. The output displays subscriber statistics specifically for the selected interface, including total sessions and their states.

```
supervisor@rtbrick.net: op> show subscriber count interface ifp-0/1/24
                       Total        Setup        Established  Terminating  Standby
ifp-0/1/24             5621         939          0            4682         0
  PPPoE                614          434          0            180          0
  L2TP                 179          0            0            179          0
  IPoE                 4828         505          0            4323         0
  L2BSA                0            0            0            0            0
  Test                 0            0            0            0            0
```

#### Subscriber Filter Show Commands

The `show subscriber filter` commands provide subscriber information using specific criteria such as interface, VLAN, MAC address, username, and session attributes. It allows you to quickly identify

and analyze subscriber sessions.

**Syntax:**

```
show subscriber filter agent-circuit-id-regex <agent-circuit-id-regex>
show subscriber filter agent-remote-id-regex <agent-remote-id-regex>
show subscriber filter count
show subscriber filter detail
show subscriber filter inner-vlan [1 - 4094]
show subscriber filter interface <interface>
show subscriber filter interface-regex <interface-regex>
show subscriber filter mac <mac>
show subscriber filter mac-regex <mac-regex>
show subscriber filter outer-vlan [1 - 4094]
show subscriber filter redundancy-session-id <rd-session-id>
show subscriber filter state [INIT | AUTHENTICATING | ADDRESS_ALLOCATION | TUNNEL_SETUP | IFL_SETUP | FULL | ACCOUNTING | ESTABLISHED | TERMINATING | STANDBY | ESTABLISHED_NO_FWD | STANDBY_FWD]
show subscriber filter type [PPPoE | L2TP | IPoE | L2BSA]
show subscriber filter username-regex <username-regex>
```

The following table provides the command options and descriptions.

| Attribute | Description |
|---|---|
| `<agent-circuit-id-regex>` | Filters subscribers based on a regular expression match of the agent circuit ID. |
| `<agent-remote-id-regex>` | Filters subscribers based on a regular expression match of the agent remote ID. |
| `count` | Displays the total number of subscribers matching the applied filter criteria. |
| `detail` | Displays detailed information for subscribers matching the applied filter criteria. |
| `inner-vlan` | Filters subscribers based on the specified inner VLAN ID. Range 1 - 4094. |
| `<interface>` | Filters subscribers associated with the specified interface. |
| `<interface-regex>` | Filters subscribers whose interface names match the specified regular expression. |
| `<mac>` | Filters subscribers based on the specified MAC address. |
| `<mac-regex>` | Filters subscribers whose MAC addresses match the specified regular expression. |
| `outer-vlan` | Filters subscribers based on the specified outer VLAN ID. Range [1 - 4094]. |
| `<rd-session-id>` | Filters subscribers based on the specified redundancy session ID. |

| Attribute | Description |
|---|---|
| `state` | Filters subscribers based on their current operational state. Supported options include INIT, AUTHENTICATING, ADDRESS_ALLOCATION, TUNNEL_SETUP, IFL_SETUP, FULL, ACCOUNTING, ESTABLISHED, TERMINATING, STANDBY, ESTABLISHED_NO_FWD, and STANDBY_FWD. |
| `type` | Filters subscribers based on the access type. Supported values are PPPoE, L2TP, IPoE, and L2BSA. |
| `<username-regex>` | Filters subscribers whose usernames match the specified regular expression. |

The following example displays a filtered list of subscribers that are in the ESTABLISHED state, including their interface, VLAN, and session type details.

```
supervisor@rtbrick.net: op> show subscriber filter state ESTABLISHED
Subscriber-Id           Interface         VLAN       Type    State
72339069014638594       ifp-0/0/0      1000:100   PPPoE   ESTABLISHED
72339069014638595       ifp-0/0/0      1001:100   PPPoE   ESTABLISHED
72339069014638596       ifp-0/0/0      100:0      PPPoE   ESTABLISHED
72339069014638597       ifp-0/0/0      1100:0     PPPoE   ESTABLISHED
72339069014638598       ifp-0/0/1      1000:100   PPPoE   ESTABLISHED
72339069014638599       ifp-0/0/1      1001:100   PPPoE   ESTABLISHED
72339069014638600       ifp-0/0/1      100:0      PPPoE   ESTABLISHED
72339069014638601       ifp-0/0/1      1100:0     PPPoE   ESTABLISHED
216454257090494465      ifp-0/0/0      1001:500   IPoE    ESTABLISHED
216454257090494466      ifp-0/0/0      500:0      IPoE    ESTABLISHED
216454257090494467      ifp-0/0/0      1500:0     IPoE    ESTABLISHED
216454257090494468      ifp-0/0/1      1000:500   IPoE    ESTABLISHED
216454257090494469      ifp-0/0/1      1001:500   IPoE    ESTABLISHED
216454257090494470      ifp-0/0/1      500:0      IPoE    ESTABLISHED
216454257090494471      ifp-0/0/1      1500:0     IPoE    ESTABLISHED
216454257090494472      ifp-0/0/0      1000:500   IPoE    ESTABLISHED
```

#### Subscriber History Show Commands

The show subscriber history commands provides details about past subscriber sessions, including the events, previously established states and so on. It allows you to query path records based on options such as interface, VLAN, MAC address, username, and so on.

**Syntax:**

**show subscriber history**
**show subscriber history** <subscriber-id>
**show subscriber history filter agent-circuit-id-regex** <agent-circuit-id-regex>
**show subscriber history filter agent-remote-id-regex** <agent-remote-id-regex>
**show subscriber history filter count**
**show subscriber history filter inner-vlan** <1-4094>
**show subscriber history filter interface** <interface>

**show subscriber history filter interface-regex** <interface-regex>
**show subscriber history filter mac** <mac>
**show subscriber history filter mac-regex** <mac-regex>
**show subscriber history filter outer-vlan** <1-4094>
**show subscriber history filter type** <PPPoE | L2TP | IPoE | Test | L2BSA>
**show subscriber history filter username-regex** <username-regex>

The following table provides command options and descriptions.

| Attribute | Description |
|---|---|
| `history` | Displays historical information for all subscriber sessions, including past states and lifecycle events. |
| `history <subscriber-id>` | Displays historical details for a specific subscriber. |
| `<agent-circuit-id-regex>` | Filters subscriber history based on a regular expression match of the agent circuit ID. |
| `<agent-remote-id-regex>` | Filters subscriber history based on a regular expression match of the agent remote ID. |
| `count` | Displays the total number of historical subscriber entries for the applied filter criteria. |
| `inner-vlan` | Filters subscriber history based on the specified inner VLAN ID. Range 1 - 4094. |
| `<interface>` | Filters subscriber history related to the specified interface. |
| `<interface-regex>` | Filters subscriber history whose interface names match the specified regular expression. |
| `<mac>` | Filters subscriber history based on the specified MAC address. |
| `<mac-regex>` | Filters subscriber history whose MAC addresses match the specified regular expression. |
| `outer-vlan` | Filters subscriber history based on the specified outer VLAN ID. Range 1 - 4094. |
| `type` | Filters subscriber history based on the access type. Supported values are PPPoE, L2TP, IPoE, and L2BSA. |
| `<username-regex>` | Filters subscriber history whose usernames match the specified regular expression. |

The following command shows termination codes that provides the reasons why subscribers are terminated for the last 24 hours and up to 4000 subscribers.

```
supervisor@rtbrick: op> show l2tp session history
Subscriber-Id           Local TID  Local SID  Terminate Code
72339069014638612            2911      27258  L2TP Tunnel Down
72339069014638610            9844      12077  L2TP Tunnel Down
```

```
72339069014638609            9844      29446  L2TP Tunnel Down
72339069014638611            9844      39942  L2TP Tunnel Down
72339069014638613           48786      64050  L2TP Tunnel Down
72339069014638614            2911      49382  Clear Session Force
72339069014638615            9844       9621  Clear Session Force
72339069014638616            9844      11781  Clear Session Force
72339069014638617            9844      20058  Clear Session Force
72339069014638618           48786      57267  Clear Session Force
```

#### Subscriber Pool Show Commands

The `show subscriber pool` commands provide information about IPv4 and IPv6 address pool utilization and allocation details. It helps you to monitor pool usage, address assignment, and analyze address allocation statistics.

**Syntax:**

**show subscriber pool ipv4** <pool>
**show subscriber pool ipv4** <pool> **allocation**
**show subscriber pool ipv6** <pool>
**show subscriber pool ipv6** <pool> **allocation**
**show subscriber pool summary**

The following table provides command options and descriptions.

| Attribute | Description |
|---|---|
| `ipv4 <pool>` | Displays information about the specified IPv4 address pool. |
| `ipv4 <pool> allocation` | Displays allocation information for the specified IPv4 pool. |
| `ipv6 <pool>` | Displays information about the specified IPv6 address pool. |
| `ipv6 <pool> allocation` | Displays allocation information for the specified IPv6 pool. |
| `summary` | Displays a summary of all IPv4 and IPv6 subscriber pools. |

The usage of local address pools can be monitored using the `show subscriber pool` commands as shown in the following example.

Rather than using recommended IP addresses for technical documents, the document shows actual IP pool ranges.

```
supervisor@rtbrick: op> show subscriber pool summary
Pool Name                        AFI  Usage           Range
IPv4_ADDRESS_POOL                IPv4 0/10            198.51.100.101 - 198.51.100.110
IPv4_ADDRESS_POOL_VRF            IPv4 0/10            198.51.100.111 - 198.51.100.120
IPv6_ADDRESS_POOL                IPv6 0/16            2001:db8:101:101::/64 -
2001:db8:101:110::/64
IPv6_PD_ADDRESS_POOL             IPv6 0/3841          2001:db8:201::/56 -
```

```
2001:db8:210::/56
```

The following example output displays the IPv4 address pool, including its address range, allocation usage, and the next linked pool for overflow.

```
supervisor@rtbrick: op> show subscriber pool ipv4 IPv4_ADDRESS_POOL
Pool Name: IPv4_ADDRESS_POOL
    AFI: IPv4
    Usage: 0/10
    Range: 198.51.100.101 - 198.51.100.110
```

The following example output displays the IPv4 address pool, showing its address range, current allocation usage, and the next pool in the chaining sequence.

```
supervisor@rtbrick: op> show subscriber pool ipv4 IPv4_ADDRESS_POOL_VRF
Pool Name: IPv4_ADDRESS_POOL_VRF
    AFI: IPv4
    Usage: 0/10
    Range: 198.51.100.111 - 198.51.100.120
```

The following example output lists detailed allocation entries for pool-B, including subscriber IDs, allocation timestamps, and the assigned IP addresses.

```
supervisor@rtbrick: op> show subscriber pool ipv4 IPv4_ADDRESS_POOL allocation
Subscriber-Id          Timestamp                            Address/Prefix
72339069014638606      Tue Jul 14 18:09:03 GMT +0000 2026   198.51.100.101
72339069014638605      Tue Jul 14 18:09:03 GMT +0000 2026   198.51.100.102
72339069014638604      Tue Jul 14 18:09:03 GMT +0000 2026   198.51.100.103
72339069014638608      Tue Jul 14 18:09:03 GMT +0000 2026   198.51.100.104
72339069014638607      Tue Jul 14 18:09:03 GMT +0000 2026   198.51.100.105
```

The following example output shows information for an IPv6 subscriber pool.

```
supervisor@rtbrick.net: op> show subscriber pool ipv6 ipoe_v6_pool3
Pool Name: ipoe_v6_pool3
    AFI: IPv6
    Usage: 0/40
    Range: fc65:1:1:b::/64 - fc65:1:1:32::/64
```

#### Subscriber Redundancy Show Commands

The `show subscriber redundancy` commands provide information about subscriber distribution across redundancy peers, assignment, failover behavior, and event history. These commands enable you to monitor redundancy states.

**Syntax:**

```
show subscriber redundancy distribution filter assignment [local | remote]
show subscriber redundancy distribution filter count
show subscriber redundancy distribution filter interface <interface>
show subscriber redundancy distribution filter lag <lag>
show subscriber redundancy distribution filter redundancy-session-id <rd-session-id>
show subscriber redundancy distribution filter subscriber-id <subscriber-id>
show subscriber redundancy distribution history redundancy-session-id <rd-session-id>
show subscriber redundancy distribution history subscriber-id <subscriber-id>
show subscriber redundancy distribution summary
show subscriber redundancy events redundancy-session-id <redundancy-session-id>
show subscriber redundancy events subscriber-id <subscriber-id>
```

| Attribute | Description |
|---|---|
| `distribution` | Displays subscriber distribution across redundancy peers. |
| `[local \| remote]` | Filters distribution output based on whether the subscriber is locally or remotely assigned. |
| `count` | Displays the total number of subscribers matching the applied filter criteria. |
| `<interface>` | Filters subscriber distribution based on the specified interface. |
| `<lag>` | Filters subscriber distribution based on the specified LAG interface. |
| `<rd-session-id>` | Filters distribution entries based on the redundancy session ID. |
| `<subscriber-id>` | Filters distribution entries based on the subscriber ID. |
| `history` | Displays distribution history of subscribers across redundancy peers. |
| `<rd-session-id>` | Displays distribution details for the specified redundancy session ID. |
| `<subscriber-id>` | Displays distribution details for the specified subscriber ID. |
| `summary` | Displays a summary of subscriber distribution across redundancy peers. |
| `events` | Displays redundancy-related information such as failover, switchover, and so on. |
| `<redundancy-session-id>` | Displays subscriber redundancy events for the specified redundancy session ID. |
| `<subscriber-id>` | Displays redundancy events for the specified subscriber ID. |

### IPoE Show Commands

The IPoE show commands provide details of IPoE subscriber sessions, redundancy, subscriber state, count, and so on. The following diagram depicts a structured or hierarchical view of the operational commands available for IPoE subscribers.

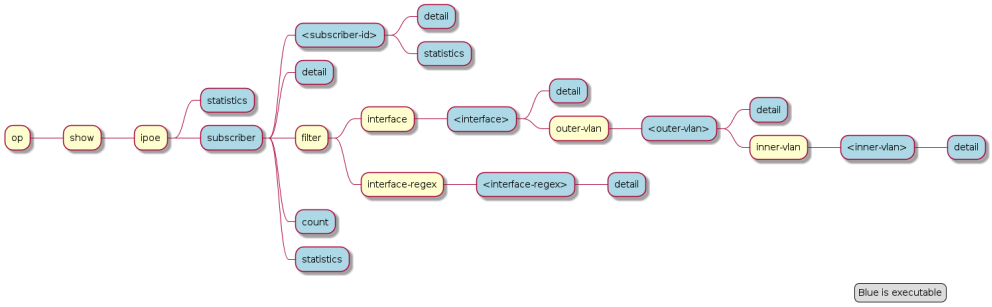



*Figure 76. IPoE Operational Commands*

#### IPoE Subscriber Show Commands

The IPoE subscriber show commands provide summary and detailed information about subscriber sessions, including total counts, per-subscriber details, errors, and traffic statistics.

```
show ipoe subscriber
show ipoe subscriber count
show ipoe subscriber detail
show ipoe subscriber errors
show ipoe subscriber statistics
```

| Option | Description |
|---|---|
| `subscriber` | Displays a summary of all IPoE subscribers. |
| `count` | Displays the total number of IPoE subscribers. |
| `detail` | Displays detailed information for all IPoE subscribers. |
| `errors` | Displays error statistics related to IPoE subscribers. |
| `subscriber statistics` | Displays aggregated statistics for all IPoE subscribers. |

#### IPoE Subscriber Details for Specific Subscriber

The IPoE subscriber show commands provide summary and detailed information traffic statistics details for a specified subscriber.

### Syntax

```
show ipoe subscriber <subscriber-id>
show ipoe subscriber <subscriber-id> detail
show ipoe subscriber <subscriber-id> statistics
```

The following table provides command options and descriptions.

| Option | Description |
|---|---|
| `subscriber <subscriber-id>` | Displays information for a specific IPoE subscriber. |
| `<subscriber-id> detail` | Displays detailed information for a specific IPoE subscriber. |

| Option | Description |
|---|---|
| `<subscriber-id> statistics` | Displays statistics for a specific IPoE subscriber. |

The following output displays detailed information for a specific IPoE subscriber, including interface attributes, DHCPv4 and DHCPv6 lease details, gateway information, and subscriber state.

```
supervisor@rtbrick: op> show ipoe subscriber 216454257090494465 detail
Subscriber-Id: 216454257090494465
    State: ESTABLISHED
    Uptime: Tue Jul 14 18:10:46 GMT +0000 2026 (0:00:04.535701)
    Interface: ifp-0/1/5
    Outer VLAN: 201 (802.1q)
    Inner VLAN: 201
    Client MAC: 02:00:00:00:00:01
    Gateway Interface: lo-0/0/1/0
    Gateway Instance: default
    Gateway MAC: 5c:07:58:79:0f:06
    Gateway IPv4: 198.51.100.11/255.255.255.255
    Agent-Remote-Id: DEU.RTBRICK.1
    Agent-Circuit-Id: 0.0.0.0/0.0.0.0 eth 0:1
    Access-Profile: IPoE_ACCESS_PROFILE
    AAA-Profile: IPoE_AAA
    DHCPv4:
        Mode: server
        Server Identifier: 198.51.100.11
        State: Bound
        Address: 198.51.100.105/255.255.255.255
        Primary DNS: 203.0.113.201
        Secondary DNS: 203.0.113.202
        Lease Created: Tue Jul 14 18:10:46 GMT +0000 2026 (0:00:04.552294)
        Lease Time: 60 seconds
        Lease Expire: 56 seconds
    DHCPv6:
        Mode: server
        State: Bound
        Client DUID: 00030001020000000001
        Server DUID: 0003001b5e0758fffe790f06
        IA_NA:
            Address: 2001:db8:101:105::
            IAID: 16777216
            Active: True
        IA_PD:
            Prefix: 2001:db8:201:400::/56
            IAID: 33554432
            Active: True
        Primary DNS: 2001:db8:201:1::201
        Secondary DNS: 2001:db8:202:1::202
        Lease Created: Tue Jul 14 18:10:46 GMT +0000 2026 (0:00:04.553767)
        Lease Time (Lifetime): 60 seconds
        Lease Expire: 56 seconds
```

```
        Preferred Lifetime: 60 seconds
```

The following output displays protocol statistics for an individual IPoE subscriber session, including DHCP, DHCPv6, and ICMPv6 message counters.

```
supervisor@rtbrick.net: op> show ipoe subscriber 216454257090494515 statistics
IPoE subscriber 216454257090494515 session statistics
DHCP packets received                       : 2
DHCP discover packets received              : 1
DHCP request packets received               : 1
DHCP packets sent                           : 2
DHCP offer packets sent                     : 1
DHCP ack packets sent                       : 1
DHCPv6 packets received                     : 1
DHCPv6 solicit packets received             : 1
DHCPv6 relay-forward packets received       : 1
DHCPv6 packets sent                         : 1
DHCPv6 reply packets sent                   : 1
DHCPv6 relay-reply packets sent             : 1
ICMPv6 router solicitation received         : 1
ICMPv6 neighbor solicitation received       : 1
ICMPv6 solicited router solicitation sent   : 1
ICMPv6 neighbor advertisement sent          : 1
```

The following output provides statistics for all IPoE subscribers, including DHCP, DHCPv6, and ICMPv6.

```
supervisor@rtbrick.net: op> show ipoe subscriber statistics
IPoE global subscriber statistics
DHCP packets received                       : 170
DHCP discover packets received              : 60
DHCP request packets received               : 60
DHCP release packets received               : 50
DHCP packets sent                           : 120
DHCP offer packets sent                     : 60
DHCP ack packets sent                       : 60
DHCPv6 packets received                     : 110
DHCPv6 solicit packets received             : 60
DHCPv6 release packets received             : 50
DHCPv6 relay-forward packets received       : 110
DHCPv6 packets sent                         : 110
DHCPv6 reply packets sent                   : 110
DHCPv6 relay-reply packets sent             : 110
ICMPv6 router solicitation received         : 60
ICMPv6 neighbor solicitation received       : 60
ICMPv6 solicited router solicitation sent   : 60
ICMPv6 neighbor advertisement sent          : 60
```

The following output a summary of all IPoE subscribers, including interface, VLAN, MAC, subscriber stat details, and DHCPv4/DHCPv6 status.

```
supervisor@rtbrick: op> show ipoe subscriber detail
Subscriber-Id           Interface         VLAN       MAC                State
DHCPv4      DHCPv6
216454257090494467      ifp-0/1/5         201:203    02:00:00:00:00:03  ESTABLISHED
Bound       Bound
216454257090494469      ifp-0/1/5         201:205    02:00:00:00:00:05  ESTABLISHED
Bound       Bound
216454257090494468      ifp-0/1/5         201:204    02:00:00:00:00:04  ESTABLISHED
Bound       Bound
216454257090494466      ifp-0/1/5         201:202    02:00:00:00:00:02  ESTABLISHED
Bound       Bound
216454257090494465      ifp-0/1/5         201:201    02:00:00:00:00:01  ESTABLISHED
Bound       Bound
```

The following output displays error and exception counters related to IPoE subscriber processing.

```
supervisor@rtbrick.net: op> show ipoe subscriber errors
No config found             : 0
No IFL found                : 0
No gateway address found    : 0
No subscriber found         : 10
Subscriber limit reached    : 0
Interface get failure       : 0
Interface down              : 0
Internal error              : 0
Subscriber not up           : 200
No chassis-id config found  : 0
Invalid redundancy state    : 0
VLAN encapsulation mismatch : 0
Relay server not configured : 0
Dropped min elapsed         : 0
```

#### IPoE Subscriber Filter Show Commands

You can view IPoE subscriber details using criteria such as interface, VLAN, MAC address, session state, and redundancy identifiers.

**show ipoe subscriber filter count**
**show ipoe subscriber filter detail**
**show ipoe subscriber filter inner-vlan** [1 - 4094]
**show ipoe subscriber filter interface** <interface>
**show ipoe subscriber filter interface-regex** <interface-regex>
**show ipoe subscriber filter mac** <mac>
**show ipoe subscriber filter mac-regex** <mac-regex>

**show ipoe subscriber filter outer-vlan** [1 - 4094]
**show ipoe subscriber filter redundancy-session-id** <rd-session-id>
**show ipoe subscriber filter state** [INIT | WAIT | ESTABLISHED | TERMINATING | TERMINATED | STANDBY]

The following table provides command options and descriptions.

| Option | Description |
|---|---|
| `count` | Displays the number of subscribers matching the applied filter criteria. |
| `detail` | Displays detailed information for subscribers matching the applied filter criteria. |
| `inner-vlan` | Displays subscribers matching the specified inner VLAN ID. VLAN ID range: 1 - 4094. |
| `<interface>` | Displays subscribers associated with the specified interface. |
| `<interface-regex>` | Displays subscribers matching the specified interface regular expression. |
| `<mac>` | Displays subscribers matching the specified MAC address. |
| `<mac-regex>` | Displays subscribers matching the specified MAC address pattern. |
| `outer-vlan` | Displays subscribers matching the specified outer VLAN ID. VLAN ID range: 1 - 4094. |
| `<rd-session-id>` | Displays subscribers matching the specified redundancy session ID. |
| `state` | Displays information a for the specified state of the subscriber. Supported values include INIT, WAIT, ESTABLISHED, TERMINATING, TERMINATED, and STANDBY` |
| `statistics` | Displays aggregated statistics for all IPoE subscribers. |
| `<subscriber-id>` | Displays information for a specific IPoE subscriber. |
| `<subscriber-id> detail` | Displays detailed information for a specific IPoE subscriber. |
| `<subscriber-id> statistics` | Displays statistics for a specific IPoE subscriber. |

The following example shows the detailed view for IPoE subscribers.

```
supervisor@rtbrick: op> show ipoe subscriber detail
Subscriber-Id           Interface         VLAN       MAC                State
DHCPv4      DHCPv6
216454257090494467      ifp-0/1/5         201:203    02:00:00:00:00:03  ESTABLISHED
Bound       Bound
216454257090494469      ifp-0/1/5         201:205    02:00:00:00:00:05  ESTABLISHED
Bound       Bound
216454257090494468      ifp-0/1/5         201:204    02:00:00:00:00:04  ESTABLISHED
```

```
Bound       Bound
216454257090494466      ifp-0/1/5         201:202    02:00:00:00:00:02  ESTABLISHED
Bound       Bound
216454257090494465      ifp-0/1/5         201:201    02:00:00:00:00:01  ESTABLISHED
Bound       Bound
```

The following example shows the detailed per subscriber information.

```
supervisor@rtbrick: op> show ipoe subscriber 216454257090494465 detail
Subscriber-Id: 216454257090494465
    State: ESTABLISHED
    Uptime: Tue Jul 14 18:10:46 GMT +0000 2026 (0:00:04.535701)
    Interface: ifp-0/1/5
    Outer VLAN: 201 (802.1q)
    Inner VLAN: 201
    Client MAC: 02:00:00:00:00:01
    Gateway Interface: lo-0/0/1/0
    Gateway Instance: default
    Gateway MAC: 5c:07:58:79:0f:06
    Gateway IPv4: 198.51.100.11/255.255.255.255
    Agent-Remote-Id: DEU.RTBRICK.1
    Agent-Circuit-Id: 0.0.0.0/0.0.0.0 eth 0:1
    Access-Profile: IPoE_ACCESS_PROFILE
    AAA-Profile: IPoE_AAA
    DHCPv4:
        Mode: server
        Server Identifier: 198.51.100.11
        State: Bound
        Address: 198.51.100.105/255.255.255.255
        Primary DNS: 203.0.113.201
        Secondary DNS: 203.0.113.202
        Lease Created: Tue Jul 14 18:10:46 GMT +0000 2026 (0:00:04.552294)
        Lease Time: 60 seconds
        Lease Expire: 56 seconds
    DHCPv6:
        Mode: server
        State: Bound
        Client DUID: 00030001020000000001
        Server DUID: 0003001b5e0758fffe790f06
        IA_NA:
            Address: 2001:db8:101:105::
            IAID: 16777216
            Active: True
        IA_PD:
            Prefix: 2001:db8:201:400::/56
            IAID: 33554432
            Active: True
        Primary DNS: 2001:db8:201:1::201
        Secondary DNS: 2001:db8:202:1::202
        Lease Created: Tue Jul 14 18:10:46 GMT +0000 2026 (0:00:04.553767)
```

```
        Lease Time (Lifetime): 60 seconds
        Lease Expire: 56 seconds
        Preferred Lifetime: 60 seconds
```

### PPPoE Show Commands

This section describes the operational commands for PPPoE sessions. You can view PPPoE session states, session establishment details, and detailed protocol-level information such as PPP negotiation, authentication, and so on.

This diagram provides the hierarchical view of the operational commands specific to PPPoE sessions.

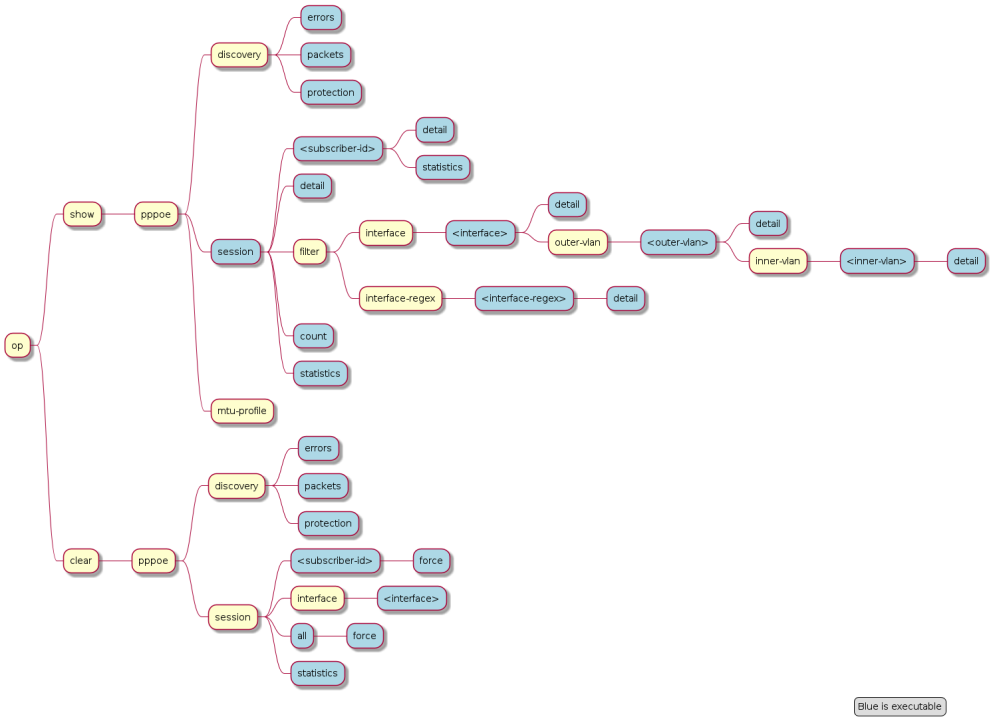



*Figure 77. PPPoE Operational Commands*

#### PPPoE Session Show Commands

These show commands provide information about PPPoE discovery, session state, protection mechanisms, and MTU profiles. You can view session details, filtering of sessions based on various parameters for all PPPoE sessions, including statistics and error conditions.

#### PPPoE Session Details

The show commands display summary information and total count of active PPPoE sessions.

**Syntax:**

```
show pppoe session
show pppoe session count
```

The following table provides command options and descriptions.

| Attribute | Description |
|---|---|
| `<cr>` | Without any option, it displays a summary of all PPPoE sessions. |
| `count` | Displays the total number of PPPoE sessions. |

The following output provides a summary of all PPPoE subscriber sessions, showing interface details, VLAN information, client MAC addresses, and the current session state.

```
supervisor@rtbrick: op> show pppoe session
Subscriber-Id           Interface         VLAN       MAC                State
72339069014638599       ifp-0/1/5         201:201    02:00:00:00:00:01  ESTABLISHED
72339069014638600       ifp-0/1/5         201:202    02:00:00:00:00:02  ESTABLISHED
72339069014638601       ifp-0/1/5         201:203    02:00:00:00:00:03  ESTABLISHED
72339069014638602       ifp-0/1/5         201:204    02:00:00:00:00:04  ESTABLISHED
72339069014638603       ifp-0/1/5         201:205    02:00:00:00:00:05  ESTABLISHED
```

The following output provides a summary of all PPPoE subscriber sessions for specified subscriber ID.

```
supervisor@rtbrick: op> show pppoe session 72339069014638599
Subscriber-Id: 72339069014638599
    State: ESTABLISHED
    Uptime: Tue Jul 14 18:08:30 GMT +0000 2026 (0:00:12.633818)
    Interface: ifp-0/1/5
    Outer VLAN: 201 (802.1q)
    Inner VLAN: 201
    Client MAC: 02:00:00:00:00:01
    Server MAC: 5c:07:58:79:0f:06
    Session-Id: 6
    Access-Profile: PPPoE_ACCESS_PROFILE
    AAA-Profile: PPPoE_AAA
```

The following output displays detailed information for a specific PPPoE subscriber session, including VLANs, authentication details, negotiated PPP parameters, and access profiles.

```
supervisor@rtbrick: op> show pppoe session 72339069014638599 detail
Subscriber-Id: 72339069014638599
    State: ESTABLISHED
    Uptime: Tue Jul 14 18:08:30 GMT +0000 2026 (0:00:13.498657)
    Interface: ifp-0/1/5
    Outer VLAN: 201 (802.1q)
    Inner VLAN: 201
    Client MAC: 02:00:00:00:00:01
    Server MAC: 5c:07:58:79:0f:06
    Session-Id: 6
    Agent-Remote-Id: DEU.RTBRICK.1
```

```
    Agent-Circuit-Id: 0.0.0.0/0.0.0.0 eth 0:1
    Access-Profile: PPPoE_ACCESS_PROFILE
    AAA-Profile: PPPoE_AAA
    PPP LCP:
        State: OPENED
        Negotiated Protocols: PAP, IPCP, IP6CP
        Negotiated Parameters: MRU, AUTH, MAGIC
        Magic Number: 2742098270 Peer: 1
        MRU: 1492 Peer: 1492
        MTU: 1492 Profile: __default_pppoe__
        Echo Interval: 10 seconds
    PAP Authentication:
        State: COMPLETED
        Username: 1@pppoe
        Reply Message: User on CBNG1
    PPP IPCP:
        State: OPENED
        Instance: default
        Peer IP Address: 198.51.100.106
        Local IP Address: 198.51.100.11
        Primary DNS: 203.0.113.201 Assigned: True
        Secondary DNS: 203.0.113.202 Assigned: True
    PPP IP6CP:
        State: OPENED
        Instance: default
        Interface Identifier: 95f6:7fd7:f30f:6ec0
        Peer Interface Identifier: 0200:00ff:fe00:0001
    IPv6:
        RA Interval: 30 seconds
        RA Prefix: 2001:db8:101:106::/64
        Delegated Prefix (DHCPv6): 2001:db8:201:500::/56 Assigned: True
        Primary DNS: 2001:db8:201:1::201
        Secondary DNS: 2001:db8:202:1::202
    Control Traffic Statistics:
        Ingress: 11 packets 819 bytes
        Egress: 11 packets 817 bytes
```

The following output displays protocol-level statistics for a PPPoE subscriber session, including PPP negotiation events and authentication message counters.

```
supervisor@rtbrick: op> show pppoe session statistics
PPPoE global session statistics
LCP configure-request received                        : 10
LCP configure-ack received                            : 10
LCP terminate-request received                        : 5
LCP terminate-ack received                            : 5
LCP echo-reply received                               : 5
LCP configure-request sent                            : 10
LCP configure-ack sent                                : 10
LCP terminate-request sent                            : 5
```

```
LCP terminate-ack sent                                : 5
LCP echo-request sent                                 : 5
LCP MRU negotiation successful                        : 10
LCP magic number negotiation successful               : 10
IPCP configure-request received                       : 20
IPCP configure-ack received                           : 10
IPCP configure-request sent                           : 10
IPCP configure-ack sent                               : 10
IPCP configure-nak sent                               : 10
IP6CP configure-request received                      : 10
IP6CP configure-ack received                          : 10
IP6CP configure-request sent                          : 10
IP6CP configure-ack sent                              : 10
PAP authentication-request received                   : 10
PAP authentication-ack sent                           : 10
ICMPv6 router-solicitation received                   : 10
ICMPv6 router-advertisement sent                      : 10
DHCPv6 solicit received                               : 10
DHCPv6 reply sent                                     : 10
```

The following output provides a summary of the total number of PPPoE sessions.

```
supervisor@rtbrick.net: op> show pppoe session count

Global:
    Total:                 10
    Established:            5
    Tunnelled:              5

Interface hostif-0/0/1:
    Total:                 10
    Established:            5
    Tunnelled:              5
```

#### Session Filter Show Commands

The show commands provide details of PPPoE sessions based on specific filter criteria such as VLANs, interface, MAC address, and session state.

**Syntax:**

**show pppoe session filter count**
**show pppoe session filter detail**
**show pppoe session filter inner-vlan** <value>
**show pppoe session filter interface** <interface>
**show pppoe session filter interface-regex** <interface-regex>
**show pppoe session filter mac** <mac>
**show pppoe session filter mac-regex** <mac-regex>

**show pppoe session filter outer-vlan** <value>
**show pppoe session filter state** [LINKING | AUTHENTICATING | NETWORKING | ESTABLISHED | TUNNELLED | TERMINATING | TERMINATED]

The following table provides command options and descriptions.

| Attribute | Description |
|---|---|
| `count` | Displays the total number of PPPoE sessions matching the applied filter criteria. |
| `detail` | Displays detailed information for PPPoE sessions matching the filter criteria. |
| `inner-vlan <value>` | Filters PPPoE sessions based on the specified inner VLAN ID. |
| `interface <interface>` | Filters PPPoE sessions associated with the specified interface. |
| `interface-regex <interface-regex>` | Filters PPPoE sessions using a regular expression match on interface names. |
| `mac <mac>` | Filters PPPoE sessions based on the specified MAC address. |
| `mac-regex <mac-regex>` | Filters PPPoE sessions using a regular expression match on MAC addresses. |
| `outer-vlan <value>` | Filters PPPoE sessions based on the specified outer VLAN ID. |
| `state [LINKING \| AUTHENTICATING \| NETWORKING \| ESTABLISHED \| TUNNELLED \| TERMINATING \| TERMINATED]` | Filters PPPoE sessions based on their current state. |

The following table provides various states and descriptions.

| State | Description |
|---|---|
| `LINKING` | PPP LCP setup. |
| `AUTHENTICATING` | PPP authentication (PAP or CHAP). |
| `NETWORKING` | PPP IPCP (IPv4) and IP6CP (IPv6) setup. |
| `ESTABLISHED` | The PPPoE session becomes established if at least one NCP (IPCP or IP6CP) is established (state OPEN). |
| `TUNNELLED` | This state indicates that a PPPoE session is tunnelled via L2TPv2. |
| `TERMINATING` | PPP session teardown. |
| `TERMINATED` | PPPoE session terminated. |

If PPPoE session remain in state TERMINATED, the subscriber state should be checked. Typically, this happens if RADIUS Accounting-Request-Stop is still pending.

#### PPPoE Subscriber Details for a Specific Subscriber

The show commands provides summary, detailed information, and statistics for PPPoE session of a specific subscriber.

**Syntax:**

> **show pppoe session** <subscriber-id>
> **show pppoe session** <subscriber-id> **detail**
> **show pppoe session** <subscriber-id> **statistics**

The following table provides command options and descriptions.

| Attribute | Description |
|---|---|
| `<subscriber-id>` | Displays a summary of the specified PPPoE session, including session state, identifiers, and associated details. |
| `<subscriber-id> detail` | Displays detailed PPPoE session information for the specified subscriber, including negotiation parameters and session attributes. |
| `<subscriber-id> statistics` | Displays statistics for the specified PPPoE subscriber, including packet and byte counters. |

#### PPPoE Session Show Commands

The show commands provides detailed information and statistics for all PPPoE sessions.

**Syntax**:

> **show pppoe session detail**
> **show pppoe session statistics**

The following table provides command options and descriptions.

| Attribute | Description |
|---|---|
| `detail` | Displays detailed information for all PPPoE sessions. |
| `statistics` | Displays protocol-level statistics for all PPPoE sessions, including packet and byte counters. |

Session statistics are available globally and per session.

Example:

```
show pppoe session statistics
show pppoe session 72339069014638601 statistics
```

The following example shows the aggregated PPPoE session control-plane statistics across all sessions on the device. The output provides counters for PPP negotiation for LCP, IPCP, and IP6CP.

```
supervisor@rtbrick: op> show pppoe session statistics
LCP configure-request received                        : 10
LCP configure-ack received                            : 10
LCP terminate-request received                        : 5
LCP terminate-ack received                            : 5
LCP echo-reply received                               : 5
LCP configure-request sent                            : 10
LCP configure-ack sent                                : 10
LCP terminate-request sent                            : 5
LCP terminate-ack sent                                : 5
LCP echo-request sent                                 : 5
LCP MRU negotiation successful                        : 10
LCP magic number negotiation successful               : 10
IPCP configure-request received                       : 20
IPCP configure-ack received                           : 10
IPCP configure-request sent                           : 10
IPCP configure-ack sent                               : 10
IPCP configure-nak sent                               : 10
IP6CP configure-request received                      : 10
IP6CP configure-ack received                          : 10
IP6CP configure-request sent                          : 10
IP6CP configure-ack sent                              : 10
PAP authentication-request received                   : 10
PAP authentication-ack sent                           : 10
ICMPv6 router-solicitation received                   : 10
ICMPv6 router-advertisement sent                      : 10
DHCPv6 solicit received                               : 10
DHCPv6 reply sent                                     : 10
```

The following example shows the PPPoE session statistics for a specific session. It displays protocol message statistics for an individual PPPoE session identified by the session ID.

```
supervisor@rtbrick: op> show pppoe session 72339069014638599 statistics
PPPoE subscriber 72339069014638599 session statistics
LCP configure-request received                        : 1
LCP configure-ack received                            : 1
LCP echo-reply received                               : 1
LCP configure-request sent                            : 1
LCP configure-ack sent                                : 1
LCP echo-request sent                                 : 1
LCP MRU negotiation successful                        : 1
LCP magic number negotiation successful               : 1
IPCP configure-request received                       : 2
IPCP configure-ack received                           : 1
IPCP configure-request sent                           : 1
IPCP configure-ack sent                               : 1
```

```
IPCP configure-nak sent                                : 1
IP6CP configure-request received                       : 1
IP6CP configure-ack received                           : 1
IP6CP configure-request sent                           : 1
IP6CP configure-ack sent                               : 1
PAP authentication-request received                    : 1
PAP authentication-ack sent                            : 1
ICMPv6 router-solicitation received                    : 1
ICMPv6 router-advertisement sent                       : 1
DHCPv6 solicit received                                : 1
DHCPv6 reply sent                                      : 1
```

For PPPoE sessions the state should be ESTABLISHED if local terminated or TUNNELLED for L2TPv2 tunnelled sessions.

```
supervisor@rtbrick: op> show pppoe session
Subscriber-Id           Interface         VLAN       MAC                 State
72339069014638599       ifp-0/1/5         201:201    02:00:00:00:00:01   ESTABLISHED
72339069014638600       ifp-0/1/5         201:202    02:00:00:00:00:02   ESTABLISHED
72339069014638601       ifp-0/1/5         201:203    02:00:00:00:00:03   ESTABLISHED
72339069014638602       ifp-0/1/5         201:204    02:00:00:00:00:04   ESTABLISHED
72339069014638603       ifp-0/1/5         201:205    02:00:00:00:00:05   ESTABLISHED
```

Alternatively, use `show pppoe session detail` which shows further details like username, Agent-Remote-Id (Line-Id) or Agent-Circuit-Id if screen width is large enough to print all those information.

The detail command shows the states of the session and all sub-protocols with extensive information and negotiated parameters.

```
supervisor@rtbrick: op> show pppoe session 72339069014638599 detail
Subscriber-Id: 72339069014638599
    State: ESTABLISHED
    Uptime: Tue Jul 14 18:08:30 GMT +0000 2026 (0:00:13.498657)
    Interface: ifp-0/1/5
    Outer VLAN: 201 (802.1q)
    Inner VLAN: 201
    Client MAC: 02:00:00:00:00:01
    Server MAC: 5c:07:58:79:0f:06
    Session-Id: 6
    Agent-Remote-Id: DEU.RTBRICK.1
    Agent-Circuit-Id: 0.0.0.0/0.0.0.0 eth 0:1
    Access-Profile: PPPoE_ACCESS_PROFILE
    AAA-Profile: PPPoE_AAA
    PPP LCP:
        State: OPENED
        Negotiated Protocols: PAP, IPCP, IP6CP
        Negotiated Parameters: MRU, AUTH, MAGIC
```

```
        Magic Number: 2742098270 Peer: 1
        MRU: 1492 Peer: 1492
        MTU: 1492 Profile: __default_pppoe__
        Echo Interval: 10 seconds
    PAP Authentication:
        State: COMPLETED
        Username: 1@pppoe
        Reply Message: User on CBNG1
    PPP IPCP:
        State: OPENED
        Instance: default
        Peer IP Address: 198.51.100.106
        Local IP Address: 198.51.100.11
        Primary DNS: 203.0.113.201 Assigned: True
        Secondary DNS: 203.0.113.202 Assigned: True
    PPP IP6CP:
        State: OPENED
        Instance: default
        Interface Identifier: 95f6:7fd7:f30f:6ec0
        Peer Interface Identifier: 0200:00ff:fe00:0001
    IPv6:
        RA Interval: 30 seconds
        RA Prefix: 2001:db8:101:106::/64
        Delegated Prefix (DHCPv6): 2001:db8:201:500::/56 Assigned: True
        Primary DNS: 2001:db8:201:1::201
        Secondary DNS: 2001:db8:202:1::202
    Control Traffic Statistics:
        Ingress: 11 packets 819 bytes
        Egress: 11 packets 817 bytes
```

#### PPPoE Discovery Show Commands

These commands provide visibility into PPPoE discovery-phase operations, including packet statistics, error conditions, and protection mechanisms applied to incoming discovery traffic.

**Syntax:**

**show pppoe discovery errors**
**show pppoe discovery packets**
**show pppoe discovery protection**
**show pppoe discovery protection filter count**
**show pppoe discovery protection filter inner-vlan** <value>
**show pppoe discovery protection filter interface** <interface-name>
**show pppoe discovery protection filter interface-regex** <interface-regex>
**show pppoe discovery protection filter mac** <mac>
**show pppoe discovery protection filter mac-regex** <mac-regex>
**show pppoe discovery protection filter outer-vlan** <value>

The following table provides command options and descriptions.

| Attribute | Description |
|---|---|
| errors | Displays PPPoE discovery error statistics. |
| packets | Displays PPPoE discovery packet statistics. |
| protection | Displays PPPoE discovery protection information and status. |
| protection filter count | Displays the total number of entries matching the PPPoE discovery protection filter criteria. |
| inner-vlan <value> | Filters PPPoE discovery protection entries based on the specified inner VLAN ID. |
| <interface-name> | Filters PPPoE discovery protection entries for the specified interface. |
| <interface-regex> | Filters PPPoE discovery protection entries using a regular expression match on interface names. |
| <mac> | Filters PPPoE discovery protection entries based on the specified MAC address. |
| <mac-regex> | Filters PPPoE discovery protection entries using a regular expression match on MAC addresses. |
| outer-vlan <value> | Filters PPPoE discovery protection entries based on the specified outer VLAN ID. |

The PPPoE discovery statistics are helpful if session setup fails in initial PPPoE tunnel setup before actual PPP negotiation is starting.

The following output provides a summary of PPPoE discovery packet activity, displaying the count of each packet type (PADI, PADO, PADR, PADS, PADT) received and sent by the system.

```
supervisor@rtbrick: op> show pppoe discovery packets
Packet           Received         Sent
PADI             10               0
PADO             0                10
PADR             10               0
PADS             0                10
PADT             5                5
```

The following output provides statistics for PPPoE discovery packet drops and errors in various conditions.

```
supervisor@rtbrick: op> show pppoe discovery errors
PADI Drop No Config            : 0
PADI Drop Session Protection   : 0
PADI Drop Session Limit        : 0
PADI Drop Outstanding Limit    : 0
PADI Drop Dup Session          : 0
```

```
PADI Drop Interface Down       : 0
PADI Drop VLAN Encap Mismatch  : 0
PADI Drop Redundancy State     : 0
PADR Drop No Config            : 0
PADR Drop Wrong MAC            : 0
PADR Drop Interface Down       : 0
PADR Drop Session Protection   : 0
PADR Drop Session Limit        : 0
PADR Drop Outstanding Limit    : 0
PADR Drop Bad Cookie           : 0
PADR Drop Bad Session          : 0
PADR Drop Duplicate Session    : 0
PADR Drop No Subscriber-Id     : 0
PADR Drop VLAN Encap Mismatch  : 0
PADR Drop Redundancy State     : 0
PADT Drop No Session           : 0
PADT Drop Wrong MAC            : 0
PADX Decode Error              : 0
PADX Internal Error            : 0
PADX Interface Get Failure     : 0
PADX Unsupported VLAN Count    : 0
PADX No chassis-id config found: 0
```

The following example provides the status of PPPoE discovery protection for subscriber interfaces and VLANs, including the number of connection attempts and the last termination reason.

```
supervisor@rtbrick: op> show pppoe discovery protection
Interface         VLAN      MAC               Status   Attempts    Last Terminate Code
ifp-0/1/5         201:201   02:00:00:00:00:01 OK       1           PPPoE LCP Terminate
Request Received
ifp-0/1/5         201:202   02:00:00:00:00:02 OK       1           PPPoE LCP Terminate
Request Received
ifp-0/1/5         201:203   02:00:00:00:00:03 OK       1           PPPoE LCP Terminate
Request Received
ifp-0/1/5         201:204   02:00:00:00:00:04 OK       1           PPPoE LCP Terminate
Request Received
ifp-0/1/5         201:205   02:00:00:00:00:05 OK       1           PPPoE LCP Terminate
Request Received
```

If PPPoE session protection is enabled in access configuration profile, short lived or failed sessions will be logged in the PPPoE session protection table (`local.pppoe.session.protection`). Every session not established for at least 60 seconds per default is considered as failed or short lived session. This will block new sessions on this IFP and VLAN's for one second per default which increase exponential with any further failed session until the max time of per default 300 seconds is reached. The interval is reset after 900 seconds without failed sessions.

The PPPoE session protection table include also last subscriber-id and terminate code which indicates the reason for session failures.

If status OK indicates that new session are accepted where BLOCKED means that sessions will be rejected.

#### PPPoE MTU Profile Show Commands

This show command displays configured PPPoE MTU profiles and their associated parameters.

**Syntax:**

```
show pppoe mtu-profile
```

The following table provides command options and descriptions.

| Attribute | Description |
|---|---|
| <cr> | Displays configured PPPoE MTU profiles and their associated parameters. |

The following output displays the configured PPPoE MTU profiles and the number of exact and best-match profile selections.

```
supervisor@rtbrick.net: op> show pppoe mtu-profile
Profile                          MTU       Exact Match       Best Match
__default_pppoe__                1492      55                0
```

### L2TP Show Commands

The L2TP operational commands provide information about the state of L2TP sessions. These commands are applicable to L2TP-based subscribers and enable you to monitor L2TP tunnels, sessions, and associated subscribers.

The following diagram shows the operational command structure and heirarchy for L2TP tunnels and sessions.

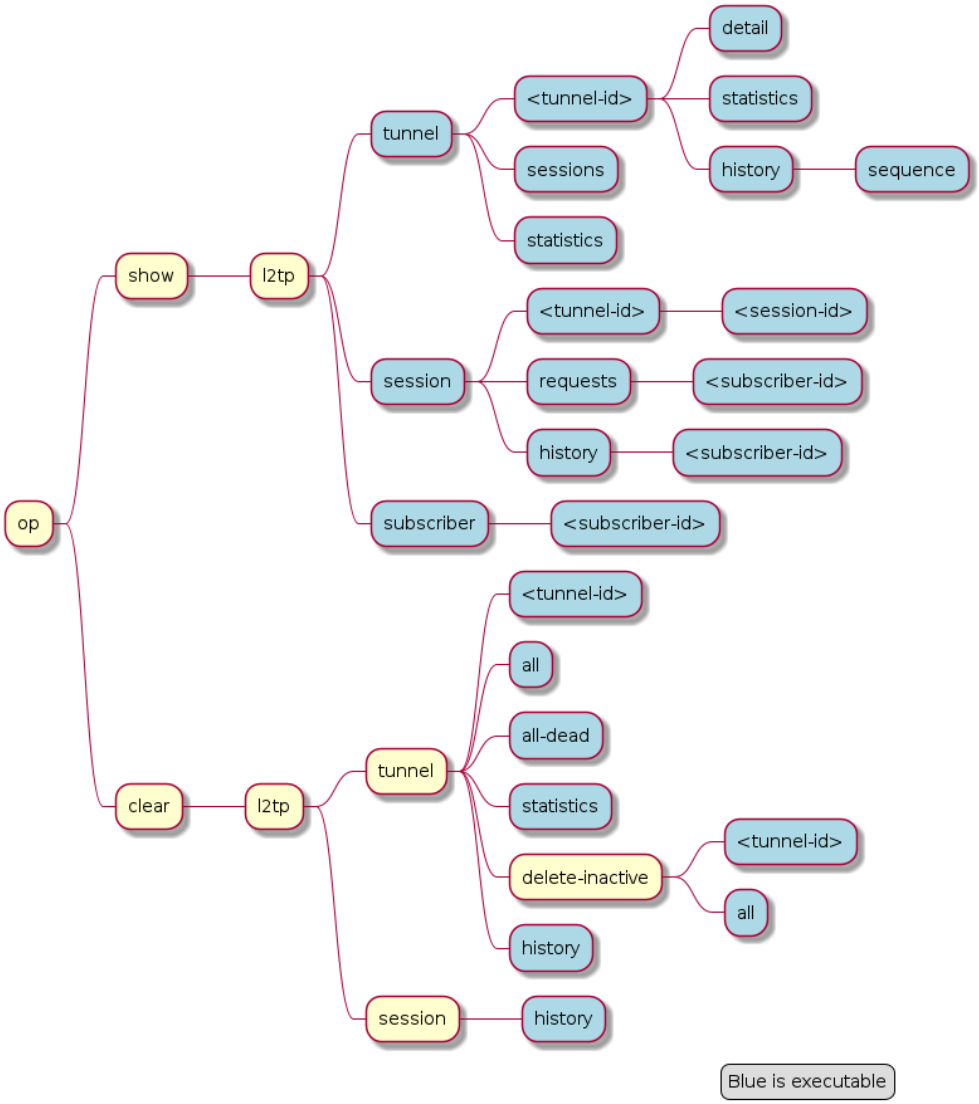



*Figure 78. L2TP Operational Commands*

#### L2TP Subscriber Show Commands

The L2TP subscriber show commands provide information about individual session using the global unique subscriber ID of L2TP subscribers.

**Syntax:**

> **show l2tp subscriber**
> **show l2tp subscriber** <subscriber-id>

The following table provides command options and descriptions.

| Option | Description |
|---|---|
| `<cr>` | Displays summary information for all L2TP subscribers. |
| `<subscriber-id>` | Displays detailed information for the specified L2TP subscriber. |

#### L2TP Tunnel

The L2TP tunnel show commands provide visibility into tunnel operations which include tunnel establishment, history, active sessions, and traffic statistics. It allows you to monitor tunnel health state.

**Syntax:**

**show l2tp tunnel**
**show l2tp tunnel history**
**show l2tp tunnel history** <sequence>
**show l2tp tunnel sessions**
**show l2tp tunnel statistics**
**show l2tp tunnel** [1 - 65535]
**show l2tp tunnel** [1 - 65535] **detail**
**show l2tp tunnel** [1 - 65535] **statistics**

The following table provides command options and descriptions.

| Option | Description |
|---|---|
| `<cr>` | Displays summary information for all active L2TP tunnels. |
| `history` | Displays historical records of L2TP tunnel events and state changes. |
| `history <sequence>` | Displays records of L2TP tunnel events for the specified sequence number. |
| `sessions` | Displays all active L2TP sessions associated with the tunnel. |
| `statistics` | Displays aggregated statistics for all L2TP tunnels. |
| `tunnel 1 - 65535` | Displays summary information for the specified L2TP tunnel ID. |
| `tunnel 1 - 65535 detail` | Displays detailed operational information for the specified L2TP tunnel ID. |
| `tunnel 1 - 65535 statistics` | Displays statistics for the specified L2TP tunnel ID. |

The following example provides information about the corresponding tunnels.

```
supervisor@rtbrick: op> show l2tp tunnel sessions
Role  Local TID  Peer TID  State        Preference  Sessions  Established  Peer Name
LAC        9844         1  ESTABLISHED       10000         3            3  LNS1
LAC       48786         2  ESTABLISHED       10000         1            1  LNS1
LAC        2911         3  ESTABLISHED       10000         1            1  LNS1
```

The following output provides a high-level summary of L2TP tunnels, total number of tunnels and the sessions successfully established on the device.

```
supervisor@rtbrick: op> show l2tp tunnel statistics
L2TP global tunnel statistics
Tunnel created              : 3
Tunnel initialized          : 3
Session created             : 5
SCCRQ sent                  : 3
SCCCN sent                  : 3
ICRQ sent                   : 5
ICCN sent                   : 5
SCCRP received              : 3
ZLB received                : 5
ICRP received               : 5
```

This following example provides the view of signaling activity specifically for L2TP tunnel '65244', detailing the control messages used to maintain that specific connection.

```
supervisor@rtbrick: op> show l2tp tunnel 9844 statistics
L2TP tunnel 9844 statistics
Session created             : 3
SCCRQ sent                  : 1
SCCCN sent                  : 1
ICRQ sent                   : 3
ICCN sent                   : 3
SCCRP received              : 1
ZLB received                : 1
ICRP received               : 3
```

#### L2TP Session Show Commands

The L2TP session show commands provide information about individual session within an L2TP tunnel which include session history and state.

**Syntax:**

**show l2tp session history** <subscriber-id>
**show l2tp session requests** <subscriber-id>
**show l2tp session** <1 - 65535>
**show l2tp session** <1 - 65535> <1 - 65535>

**show l2tp session**
**show l2tp session history**
**show l2tp session history** <subscriber-id>
**show l2tp session requests**
**show l2tp session requests** <subscriber-id>
**show l2tp session** <1-65535>

**show l2tp session** <1-65535> <1-65535>

The following table provides command options and descriptions.

| Option | Description |
|---|---|
| `<cr>` | Displays summary information for all active L2TP sessions. |
| `history` | Displays historical records of L2TP session events and state changes. |
| `history <subscriber-id>` | Displays historical L2TP session records for the specified subscriber ID. |
| `requests` | Displays pending or processed L2TP session requests. |
| `requests <subscriber-id>` | Displays L2TP session requests for the specified subscriber ID. |
| `session <1- 65535>` | Displays information for the specified L2TP session ID. |
| `<1-65535> <1-65535>` | Displays information for the specified L2TP tunnel ID and session ID combination. |

For L2TPv2 tunnelled PPPoE sessions, the globally unique subscriber ID can be used to retrieve detailed session-level information.

The following output displays a summary of all L2TP subscriber sessions currently present on the system, including the subscriber ID and the operational state of each session.

```
supervisor@rtbrick.net: op> show l2tp subscriber
Subscriber-Id           State
72339069014638658       ESTABLISHED
72339069014638657       ESTABLISHED
72339069014638654       ESTABLISHED
72339069014638655       ESTABLISHED
72339069014638656       ESTABLISHED
```

The following output displays detailed information for a specific L2TP subscriber, including tunnel and session identifiers, and connection state.

```
supervisor@rtbrick: op> show l2tp subscriber 72339069014638609
Subscriber-Id: 72339069014638609
    State: ESTABLISHED
    Local TID: 9844
    Local SID: 29446
    Peer TID: 1
    Peer SID: 2
    Call Serial Number: 1
    TX Speed: 16384000 bps
    RX Speed: 1024000 bps
```

```
    CSUN: disabled
```

The following example provides the list of L2TP session terminations with subscriber identifiers and associated tunnel IDs

```
supervisor@rtbrick: op> show l2tp session history
Subscriber-Id           Local TID  Local SID  Terminate Code
72339069014638612            2911      27258  L2TP Tunnel Down
72339069014638610            9844      12077  L2TP Tunnel Down
72339069014638609            9844      29446  L2TP Tunnel Down
72339069014638611            9844      39942  L2TP Tunnel Down
72339069014638613           48786      64050  L2TP Tunnel Down
72339069014638614            2911      49382  Clear Session Force
72339069014638615            9844       9621  Clear Session Force
72339069014638616            9844      11781  Clear Session Force
72339069014638617            9844      20058  Clear Session Force
72339069014638618           48786      57267  Clear Session Force
```

#### L2TP Result and Disconnect Codes Show Commands

The following example displays a list of previously terminated L2TP tunnels, including tunnel identifiers, timestamps, and termination code.

The received result (RFC2661) and disconnect (RFC3145) code and message from CDN and StopCCN will be stored similar to the subscriber terminate history table for 24 hours and up to 1000 records.

```
supervisor@rtbrick: op> show l2tp tunnel history
Sequence  Local TID  Peer TID  Timestamp                            Terminate Code
       1       2911         3  Tue Jul 14 18:13:32 GMT +0000 2026    Admin Request
       2       9844         1  Tue Jul 14 18:13:32 GMT +0000 2026    Admin Request
       3      48786         2  Tue Jul 14 18:13:32 GMT +0000 2026    Admin Request
```

The following example provides information about a specific tunnel history entry, including local and peer parameters.

```
supervisor@rtbrick: op> show l2tp tunnel history 3
Local TID: 48786 Peer TID: 2
    Terminate Code: Admin Request
    Timestamp: Tue Jul 14 18:13:32 GMT +0000 2026
    Local Address: 198.51.100.11
    Peer Address: 192.0.2.2
    Peer Name: LNS1
    Tunnel-Client-Auth-ID: BNG
    Tunnel-Server-Auth-ID: LNS3
```

The following example provides information about L2TP if found for the corresponding subscriber.

```
supervisor@rtbrick: op> show subscriber history
Subscriber-Id           Type    Timestamp                               Terminate Code
72339069014638594       PPPoE   Tue Jul 14 18:08:29 GMT +0000 2026    Clear Session
Force
72339069014638595       PPPoE   Tue Jul 14 18:08:29 GMT +0000 2026    Clear Session
Force
72339069014638596       PPPoE   Tue Jul 14 18:08:29 GMT +0000 2026    Clear Session
Force
72339069014638597       PPPoE   Tue Jul 14 18:08:29 GMT +0000 2026    Clear Session
Force
72339069014638598       PPPoE   Tue Jul 14 18:08:29 GMT +0000 2026    Clear Session
Force
```

The following example shows information related to L2TP tunnel and session establishment. The output includes statistics for tunnel creation, initialization, and control messages exchanged during the L2TP control-plane signaling process.

```
supervisor@rtbrick: op> show l2tp tunnel statistics
L2TP global tunnel statistics
Tunnel created              : 3
Tunnel initialized          : 3
Session created             : 5
SCCRQ sent                  : 3
SCCCN sent                  : 3
ICRQ sent                   : 5
ICCN sent                   : 5
SCCRP received              : 3
ZLB received                : 5
ICRP received               : 5
```

#### L2TP Cross-Connect Show Commands

L2TP cross-connect show commands display information about L2TP cross-connect sessions.

**Syntax:**

**show forwarding l2tp cross-connect**
**show forwarding l2tp cross-connect detail**
**show forwarding l2tp cross-connect summary**
**show forwarding l2tp cross-connect filter type (PPPoE-L2TP | L2TP-PPPoE)**
**show forwarding l2tp cross-connect filter type (PPPoE-L2TP | L2TP-PPPoE) detail**

The following table provides command options and descriptions.

| Option | Description |
|---|---|
| `<cr>` | Without any option, it displays information about all L2TP cross connect sessions. |
| `detail` | Displays detailed information for each L2TP cross-connect session. |
| `summary` | Displays a high-level summary of L2TP cross-connect sessions. |
| `type (PPPoE-L2TP \| L2TP-PPPoE)` | `PPPoE-L2TP` option displays only sessions with L2TP to PPPoE direction. `PPPoE-L2TP` option displays only sessions with PPPoE to L2TP direction. |
| `type (PPPoE-L2TP \| L2TP-PPPoE) detail` | Displays detailed information for the L2TP sessions for the specified direction. |

The following example displays statistics of L2TP cross-connect sessions. It includes total, ingress and egress, and success and failure counts.

```
supervisor@rtbrick.net: cfg> show forwarding l2tp-cross-connect summary
Total cross connects: 30
Ingress cross connects: 30
Egress cross connects: 0
Success cross connects: 30
Failed cross connects: 0
```

The following example displays displays a list of L2TP cross-connect sessions with their name, direction, type, and status.

```
supervisor@rtbrick: op> show l2tp tunnel statistics
L2TP global tunnel statistics
Tunnel created              : 3
Tunnel initialized          : 3
Session created             : 5
SCCRQ sent                  : 3
SCCCN sent                  : 3
ICRQ sent                   : 5
ICCN sent                   : 5
SCCRP received              : 3
ZLB received                : 5
ICRP received               : 5
```

```
supervisor@rtbrick.net: cfg> show forwarding l2tp-cross-connect
Name                                Direction     Type               Status
L2TP-TO-PPPOE-XC-72339069014638594    ingress       L2TP to PPPoE      Download
success
L2TP-TO-PPPOE-XC-72339069014638595    ingress       L2TP to PPPoE      Download
success
L2TP-TO-PPPOE-XC-72339069014638596    ingress       L2TP to PPPoE      Download
success
L2TP-TO-PPPOE-XC-72339069014638597    ingress       L2TP to PPPoE      Download
```

```
success
L2TP-TO-PPPOE-XC-72339069014638598    ingress       L2TP to PPPoE      Download
success
PPPOE-TO-L2TP-XC-72339069014638604    ingress       PPPoE to L2TP      Download
success
PPPOE-TO-L2TP-XC-72339069014638605    ingress       PPPoE to L2TP      Download
success
PPPOE-TO-L2TP-XC-72339069014638606    ingress       PPPoE to L2TP      Download
success
PPPOE-TO-L2TP-XC-72339069014638607    ingress       PPPoE to L2TP      Download
success
PPPOE-TO-L2TP-XC-72339069014638608    ingress       PPPoE to L2TP      Download
success
```

The following example displays detailed information for each L2TP cross-connect session, including source and destination ports along with status.

```
supervisor@rtbrick.net: cfg> show forwarding l2tp-cross-connect detail
Name                                  Direction     Type               Source Port
Destination Port    Status
L2TP-TO-PPPOE-XC-72339069014638594    ingress       L2TP to PPPoE      1276167002
2550189302          Download success
L2TP-TO-PPPOE-XC-72339069014638595    ingress       L2TP to PPPoE      1276167003
2550189304          Download success
PPPOE-TO-L2TP-XC-72339069014638600    ingress       PPPoE to L2TP      1277165588
2550189315          Download success
PPPOE-TO-L2TP-XC-72339069014638601    ingress       PPPoE to L2TP      1277165589
2550189317          Download success
PPPOE-TO-L2TP-XC-72339069014638602    ingress       PPPoE to L2TP      1277165590
2550189319          Download success
PPPOE-TO-L2TP-XC-72339069014638603    ingress       PPPoE to L2TP      1277165591
2550189321          Download success
PPPOE-TO-L2TP-XC-72339069014638604    ingress       PPPoE to L2TP      1277165592
2550189323          Download success
```

### 5.1.4. Supported Standards

RFC and draft compliance are partial except as specified.

**PPPoE**

- RFC 1516
- RFC 1661
- RFC 1332
- RFC 5072
- RFC 1334

- RFC 4638

**RADIUS**

- RFC 2865
- RFC 3162
- RFC 2866
- RFC 4372
- RFC 2869

**DHCPv4**

- RFC 951
- RFC 1542
- RFC 2131
- RFC 2132
- RFC 3046

**DHCPv6**

- RFC 8415

**Access Line Information**

The access line identification and characterization information are defined in the Broadband Forum (BBF) formerly known DSL Forum attributes including Agent-Remote-Id and Agent-Circuit-Id.

See the following references for more information about access line attributes.

- RFC 4679 DSL Forum Vendor-Specific RADIUS Attributes
- RFC 6320 ANCP
- Broadband Forum TR-101
- draft-lihawi-ancp-protocol-access-extension-04

**L2TPv2**

RFC and draft compliance are partial except as specified.

**RFC 2661 - Layer Two Tunneling Protocol (L2TPv2)**

RFC compliant L2TPv2 Access Concentrator (LAC) with the following protocol limitations:

- No support for LNS initiated outbound calls (OCRQ, OCRP and OCCN)
- No support for WAN-Error-Notify (WEN) Messages send by LAC to LNS

- No support for Set-Link-Info (SLI) Messages send by LNS to LAC
- No support for L2TP over IPv6
- No support for L2TP offset values other than 0.

**RFC 5515 - L2TP Access Line Information AVP Extensions**

- Support for access line AVP send (LAC) and received (LNS) as part of the L2TP Incoming-Call-Request (ICRQ) message.
- Response to Connect-Speed-Update-Request (CSURQ) L2TP messages is currently not supported.

**RFC 2868 - RADIUS Attributes for Tunnel Protocol Support**

RADIUS support for L2TP with the following limitations:

- No support of FQDN format for IP addresses
- No support Tunnel-Medium-Type other than IPv4

**RFC 3145 - L2TP Disconnect Cause Information**

Send meaningful disconnect cause information to LNS and display received disconnect cause information for tunnels and sessions.

**Supported Hardware**

You can find more detailed information about what RtBrick features are supported on each hardware platform, see the *Platform Guide*.

# 5.2. RBFS RADIUS Services

## 5.2.1. RADIUS Services Overview

The modular, scalable subscriber management that RtBrick calls the next generation access infrastructure (ng-access) provides support for protocols such as PPPoE, IPoE, L2TP and RADIUS.

The subscriber management infrastructure provides the next generation of internet access protocols designed for carrier grade services in regards to scalability and robustness. One of the challenges for carrier networks is interwork with numerous client devices which requires a well implemented, industry proven access protocol stack, including support for all relevant RFCs.

This implementation is designed to be a set of distributed services for increased scaling and stability. The subscriber management implementation has three components:

- subscriberd: subscriber management and AAA (local, RADIUS, and other support)
- pppoed: PPPoE/PPP session handling
- l2tpd: L2TP tunnel and session handling

The subscriber daemon (subscriberd) is the central application, keeping the current subscriber state as well as being responsible for Authentication, Authorization and Accounting (AAA).

This document describes the subscriber management RADIUS service implementation on RBFS. The term subscriber describes an access user or session from a higher level decoupled from underlying protocols like PPPoE or IPoE. Subscribers in RBFS can be managed locally or via RADIUS where this document considers RADIUS only. Each subscriber is uniquely identified by a 64bit number called subscriber-id. Remote Authentication Dial-In User Service (RADIUS) is a networking protocol that provides centralized Authentication, Authorization and Accounting (AAA) management for all types of subscribers (PPPoE, or IPoE). RADIUS servers can perform as authentication and accounting servers or change of authorization (CoA) clients. Authentication servers maintain authentication records for subscribers. The subscriber daemon requests authentication in RADIUS access-request messages before permitting subscribers access. Accounting servers handle accounting records for subscribers. The subscriber daemon transmits RADIUS accounting-start, interim and stop messages to the servers. Accounting is the process of tracking subscriber activity and network resource usage in a subscriber session. This includes the session time called time accounting and the number of packets and bytes transmitted during the session called volume accounting.

A RADIUS server can behave as a change of authorization (CoA) client allowing dynamic changes for subscriber sessions. The subscriber daemon supports both RADIUS CoA messages and disconnect messages. CoA messages can modify the characteristics of existing subscriber sessions without loss of service, disconnect messages can terminate subscriber sessions.

Each RADIUS request from subscriber daemon includes the RADIUS accounting-session-id attribute (type 44) with a format which is configurable in the AAA profile and includes at least the subscriber-id to identify the corresponding subscriber. The default format (<subscriber-id>.<timestamp>) includes also an Unix timestamp to ensure that the tuple of NAS-Identifier (e.g. hostname) and Accounting-Session-Id is global unique to be usable as key in RADIUS databases.

Additionally to subscriber-id and accounting-session-id each subscriber consists also of a subscriber-ifl build based on physical port information and subscriber-id (ifp: ifp-0/0/1 and subscriber-id: 72339069014638610 → subscriber-ifl: ppp-0/0/1/72339069014638610) which is required as handle in the RBFS forwarding infrastructure. All those three informations are part of the RADIUS access-request message:

- accounting-session-id: standard attribute 44
- subscriber-id: RtBrick VSA (26-50058-25 RtBrick-Subscriber-Id)
- subscriber-ifl: RtBrick VSA (26-50058-26 RtBrick-Subscriber-Ifl)

```
    Code: Access-Request (1)
    Packet identifier: 0x22 (34)
    Length: 416
    Authenticator: e61a0dd74c74704f608688b08de1dfba
    [The response to this request is in frame 12]
  ▼ Attribute Value Pairs
    ▶ AVP: t=User-Name(1) l=19 val=user1@rtbrick.com
    ▶ AVP: t=CHAP-Challenge(60) l=18 val=2f696f4e920b47cab869021feb2bf632
    ▶ AVP: t=CHAP-Password(3) l=19 val=02f439040e9feb7bbc9e7622a364344913
    ▶ AVP: t=NAS-IP-Address(4) l=6 val=1.1.1.1
    ▶ AVP: t=NAS-Identifier(32) l=5 val=BNG
    ▶ AVP: t=NAS-Port-Id(87) l=59 val=BNG#hostif-0/0/4#10#7#0.0.0.0/0.0.0.0 eth 1#DEU.RTBRICK.1
    ▶ AVP: t=NAS-Port(5) l=6 val=67149831
    ▶ AVP: t=NAS-Port-Type(61) l=6 val=Ethernet(15)
    ▶ AVP: t=Service-Type(6) l=6 val=Framed(2)
    ▶ AVP: t=Framed-Protocol(7) l=6 val=PPP(1)
    ▶ AVP: t=Acct-Session-Id(44) l=30 val=72339069014638895:1589876315
    ▶ AVP: t=Vendor-Specific(26) l=13 vnd=RtBrick Inc.(50058)
    ▶ AVP: t=Vendor-Specific(26) l=20 vnd=RtBrick Inc.(50058)
    ▶ AVP: t=Vendor-Specific(26) l=16 vnd=RtBrick Inc.(50058)
    ▶ AVP: t=Vendor-Specific(26) l=25 vnd=RtBrick Inc.(50058)
    ▼ AVP: t=Vendor-Specific(26) l=16 vnd=RtBrick Inc.(50058)
        Type: 26
        Length: 16
        Vendor ID: RtBrick Inc. (50058)
      ▶ VSA: t=RtBrick-Subscriber-Id(25) l=10 val=010100000000012f
    ▼ AVP: t=Vendor-Specific(26) l=35 vnd=RtBrick Inc.(50058)
        Type: 26
        Length: 35
        Vendor ID: RtBrick Inc. (50058)
      ▶ VSA: t=RtBrick-Subscriber-Ifl(26) l=29 val=ppp-0/0/4/72339069014638895
    ▶ AVP: t=Vendor-Specific(26) l=29 vnd=The Broadband Forum(3561)
    ▶ AVP: t=Calling-Station-Id(31) l=23 val=0.0.0.0/0.0.0.0 eth 1
    ▶ AVP: t=Vendor-Specific(26) l=21 vnd=The Broadband Forum(3561)
    ▶ AVP: t=Vendor-Specific(26) l=18 vnd=The Broadband Forum(3561)
```

*Figure 79. RADIUS Services Overview*

> The subscriber-id is an unsigned 64bit integer which is shown as a hex number in Wireshark.

Each subscriber is formed based on configuration profiles and individual settings retrieved via RADIUS. Conflicts between RADIUS defined attributes and profile attributes are solved by prioritizing those received from RADIUS which is common best practices for broadband access concentrators. New subscribers are signalled via RADIUS access-request and either accepted by RADIUS access-accept or rejected by RADIUS access-reject message from RADIUS server. The RADIUS access-accept includes all attributes required to form the subscriber like IP addresses, DNS servers and referenced configuration profiles. Some of those attributes can be changed by RADIUS dynamically using CoA requests without disconnecting the subscriber.

Some of those attributes refer to RADIUS services which are described in detail in this document. The term RADIUS services described the ability to dynamically control the properties of a session via RADIUS used by access providers to build their different products and services based on their network infrastructure. This is used to control QoS settings like shaper rates dynamically based on changed line quality or possible volume quotas. It is also used to dynamically enable or disable services like Voice or IPTV on a per subscriber basis.

The RADIUS accounting accuracy should comply with §45g of the German Telecommunications Act (TKG) further named by TKG§45. This applies for time and volume based products and services. Products or services with unlimited volumes but with changed properties after a certain amount of traffic are considered as volume based products as well. One example here is a fair use policy which enforces a rate limit after a certain volume. The requirements for time based accounting apply only to products or services which are charged by time or changed properties after a certain time which is not very common in the market today.

**Supported Platforms**

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

## 5.2.2. RADIUS Service Solution Overview

The general concept of RADIUS services in RtBrick FullStack (RBFS) consists of pre-configured profiles in the global static configuration assigned to dynamic subscribers via RADIUS VSA in access-accept and CoA request messages. The profiles are managed as any other configuration in RBFS via corresponding API which is not explained here in detail.

The following pictures show the concept of global static configurations referenced by RADIUS controlled dynamic subscribers.

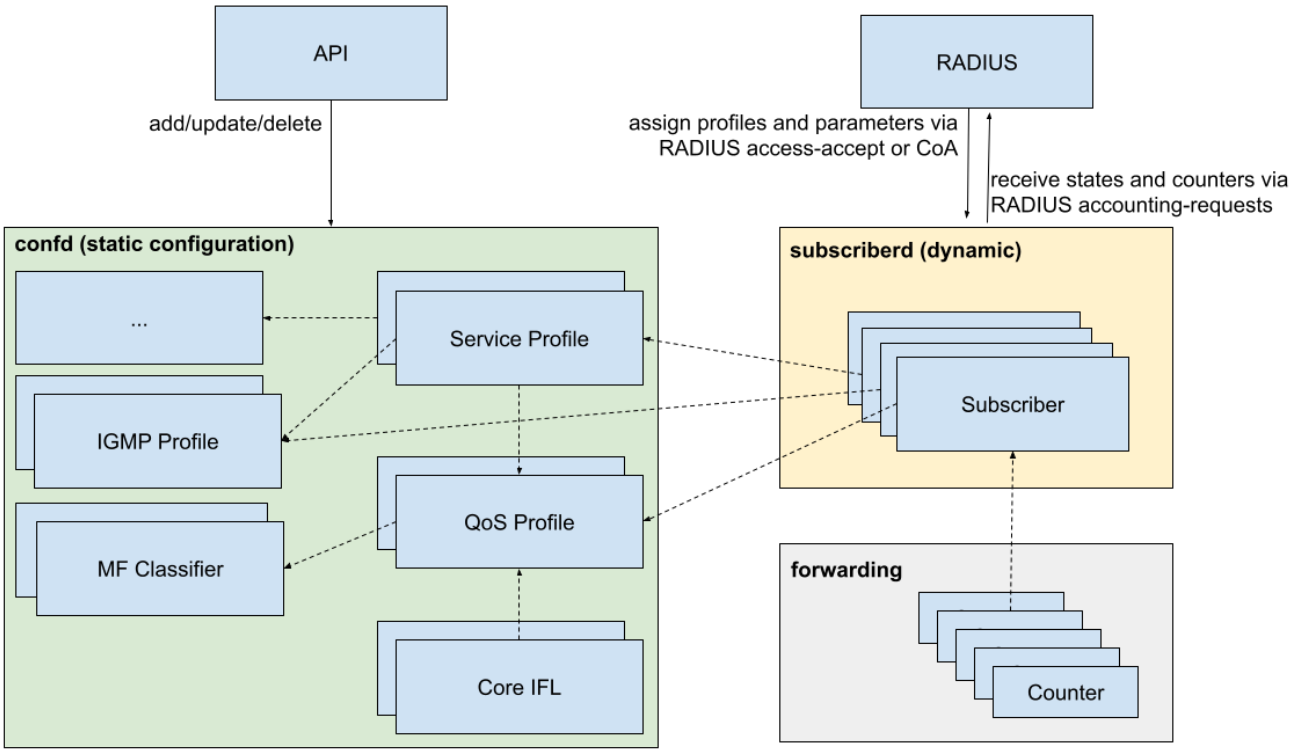



*Figure 80. RADIUS Service Solution*

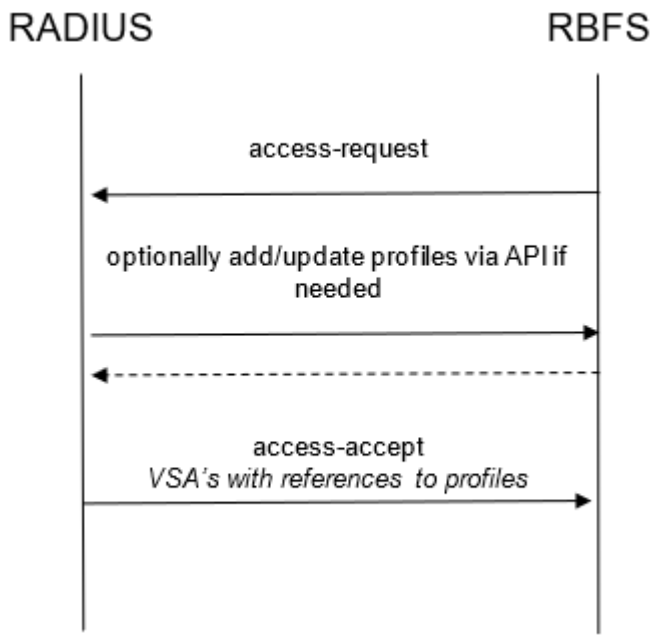



*Figure 81. RADIUS Profiles and Filters*

The required profiles and filters should be present before service becomes active which can be done immediately after start-up or between access-request and access-accept as shown in the flow diagram.

### 5.2.3. RADIUS Control

This chapter explains the subscriber and service-related RBFS extensions and RtBrick VSA related to RADIUS services.

#### RADIUS Session and Idle Timeout

The session and idle timeout values are initialized with zero, which means infinity or disabled. Those values will be optionally overwritten with the values in AAA configuration profile and RADIUS access-accept if present. The priority is RADIUS attributes over AAA configuration profile.

The session (attribute 27) and idle (attribute 28) timeout RADIUS attributes are defined in RFC 2865.

RFC and draft compliance are partial except as specified.

The idle timeout is based on outgoing logical interface statistics (OutLIF) for the subscriber-ifl to determine subscriber inactivity.

#### RADIUS Routing Instance

***VSA 26-50058-137 - RtBrick-Instance***

This attribute is applicable to non-IPoE subscribers only. It allows to change the routing instance through a RADIUS access-accept message.

For IPoE subscribers, the routing instance is automatically determined based on the gateway IFL.

**RADIUS Gateay-Ifl**

***VSA 26-50058-138 - RtBrick-Gateway-Ifl***

This attribute is applicable to IPoE subscribers only. It allows to change the actual gateway IFL. Consequently, this also indirectly modifies the routing instance through a RADIUS Access-Accept message.

**RADIUS DNS Server**

***VSA 26-50058-131 - RtBrick-DNS-Primary***

***VSA 26-50058-132 - RtBrick-DNS-Secondary***

***VSA 26-50058-134 - RtBrick-DNS-Primary-IPv6***

***VSA 26-50058-135 - RtBrick-DNS-Secondary-IPv6***

Those attributes allow to set the primary and secondary IPv4 and IPv6 DNS servers via RADIUS access-accept. DNS servers already set via access configuration profile will be overwritten by RADIUS.

**RADIUS Service Profile**

***VSA 26-50058-70 - RtBrick-Service-Profile***

Subscribers can be associated with a service-profile which defines the actual service properties like QoS or IGMP profiles and configuration settings. The service profile can be assigned or changed using the RtBrick-Service-Profile VSA.

```
RtBrick-Service-Profile = <service-profile>
```

This attribute is supported via RADIUS access-accept and CoA requests.

All dynamic QoS settings like shaper and policer rates will be reset if the new service-profile includes a qos-profile attribute also if active qos-profile and old qos-profile is equal. All QoS settings remain unchanged if the referenced service profile does not include the qos-profile attribute. If the referenced service profile updates the qos-profile attribute and additional shaper or policer rates are included in the same CoA request which updates the service-profile, those shaper and policer settings will be applied to the new QoS configuration profile after reset. This means that we reset all incremental changes done before. In example if Voice shaper rate has changed to another value, after profile change the default value from profile is used.

RADIUS CoA requests referencing a service profile which is not found on device will be rejected with CoA-NAK (invalid-request) but without changing any subscriber QoS settings. This behavior is different for RADIUS access-accept where service profiles are just ignored if not found. In both cases a warning is generated in subscriber daemon log if a referenced service profile is not found.

**RADIUS AAA Profile**

***VSA 26-50058-69 - RtBrick-AAA-Profile***

This attribute allows to change the associated AAA profile from RADIUS access-accept. This is primarily used to change the accounting adjustment values which are defined in this profile. Simple example here is to use different adjustment values for L2TP and local terminated PPPoE sessions. Another valid use case is to assign different RADIUS accounting servers for in example L2TP sessions or wholesale customers.

```
RtBrick-AAA-Profile = <aaa-profile>
```

This attribute is supported via RADIUS access-accept only.

**RADIUS QoS Profile**

***VSA 26-50058-62 - RtBrick-QoS-Profile***

```
RtBrick-QoS-Profile = <qos-profile>
```

This attribute is supported by RADIUS access-accept and CoA request. The QoS configuration profile can be either selected via service-profile or directly using this attribute which has priority of the service-profile.

All dynamic QoS settings like MFC, queue sizes, shaper and policer rates will be reset if this attribute is present in a CoA request also if new qos-profile and old qos-profile is equal. If additional shaper or policer rates are included in the same CoA request which updates the service-profile, those shaper and policer settings will be applied to the new QoS configuration after reset.

The subscriber management infrastructure does not check if a referenced QoS profile exists or not. This is handled by forwarding infrastructure which continues processing the subscriber QoS settings as soon as the profile was added. In the meantime there is no QoS configuration applied.

**RADIUS QoS Parent Scheduler**

***VSA 26-50058-64 - RtBrick-QoS-Parent-Scheduler***

The parent scheduler element of the scheduler-map assigned to the subscriber can be selected with this attribute. If not present, the scheduler-map will be directly bound to the local IFP where the session is established.

This attribute is supported in RADIUS access-accept only (no CoA support) and will set the parent scheduler element of the subscriber.

> Providing a QoS parent scheduler which is not present on the corresponding IFP will lead to black howling of all egress data traffic. Control traffic is not impacted and therefore the session will remain.

**RADIUS QoS Shaper**

***VSA 26-50058-63 - RtBrick-QoS-Shaper***

Subscribers can be associated with a QoS profile which is assigned via service-profile or directly via corresponding VSA (RtBrick-QoS-Profile). This profile is used to instantiate the QoS resources for

the subscriber including schedulers, queues and shapers. The assigned shaper instances can be updated using the RtBrick-QoS-Shaper VSA (attribute 26-50058-63) which will apply to the QoS instance of the corresponding subscriber only, but not to the other subscribers using the same QoS profile. It is only possible to update existing shapers dynamically but it is not possible to create a new shaper via RADIUS.

The RtBrick-QoS-Shaper value is a string which contains a list of multiple shaper settings separated by semicolon. Each shaper setting contains a shaper name, high flow rate and low flow rate separated by comma. The actual shaper rate is the sum of high and low flow rate.

```
RtBrick-QoS-Shaper = <shaper-name>,<high-kbps>,<low-kbps>;<shaper-name>,...
```

This attribute can be also used as a key-value list which is automatically recognized by RBFS.

```
RtBrick-QoS-Shaper = name=<shaper-name>,high=<high-kbps>,low=<low-kbps>;...
```

This attribute is supported by RADIUS access-accept and CoA request.

Updating a single shaper (e.g. voice_shaper) via CoA does not require to include other shapers meaning that only the shapers included in the attribute will be updated and all other shapers remain unchanged.

Assume the following example which adapts the session and voice shaper instance of a subscriber.

```
supervisor@rtbrick: op> show config forwarding-options class-of-service shaper
shaper_session
{
  "rtbrick-config:shaper": {
    "shaper-name": "shaper_session",
    "shaping-rate-high": 48000,
    "shaping-rate-low": 2000
  }
}
supervisor@rtbrick: op> show config forwarding-options class-of-service shaper
shaper_voice
{
  "rtbrick-config:shaper": {
    "shaper-name": "shaper_voice",
    "shaping-rate-high": 1000,
    "shaping-rate-low": 0
  }
}
```

**RADIUS VSA Change Session Shaper Only**

```
RtBrick-QoS-Shaper: shaper_session,14000,2000
```

**RADIUS VSA Change Session and Voice Shaper**

```
RtBrick-QoS-Shaper: shaper_session,14000,2000;shaper_voice,2000
```

All active dynamic shapers are stored in the table **global.access.1.subscriber.qos.shaper** to handover those information to forwarding infrastructure. This table can be used to verify the dynamic shapers which are active for a given subscriber.

The CLI command `show subscriber <id> qos` displays those information in a nice human readable format.

```
supervisor@rtbrick: op> show subscriber 72339069014638610 qos
Subscriber-Id: 72339069014638610
    Interface: ifp-0/0/1
    Outer VLAN: 128
    Inner VLAN: 7
    IFL: ppp-0/0/1/72339069014638610
    Profile: qos_profile
    Parent: pon1
    Dynamic Shaper: shaper_voice
        Rate Low: 0 kbps
        Rate High: 2000 kbps
    Dynamic Shaper: shaper_session
        Rate Low: 2000 kbps
        Rate High: 14000 kbps
```

A shaper rate of 0 means infinity!

**RADIUS QoS Policer**

***VSA 26-50058-65 - RtBrick-QoS-Policer***

The RtBrick-QoS-Policer value is a string which contains a list of multiple policer level settings separated by semicolon. Each setting contains a level, cir, cbs, pir, pbs, max-cir and max-pir separated by comma.

RtBrick-QoS-Policer = <level>,<cir>,<cbs>,<pir>,<pbs>,<max-cir>,<max-pir>;<level>...

*Example:*
RtBrick-QoS-Policer = 1,2000,200;2,8000,800;3,0,800;4,0,800

- **level**: 1 (highest priority) to 4 (lowest priority)
- **cir**: Ingress policer committed information rate (kbps)
- **cbs**: Ingress policer committed burst size (kbits)
- **pir**: Ingress policer peak information rate (kbps)
- **pbs**: Ingress policer peak burst size (kbits)
- **max-cir**: max ingress policer committed information rate (kbps)

- **max-pir**: max ingress policer peak information rate (kbps)

If PIR and PBS are not defined, the values from CIR and CBS are used as PIR and PBS as well. The max CIR and max PIR attributes are optional default set to unlimited.

This attribute can be also used as a key-value list which is automatically recognized by RBFS.

RtBrick-QoS-Policer = level=<level>,cir=<cir>,cbs=<cbs>,pir=<pir>,pbs=<pbs>,max-cir=<max-cir>,max-pir=<max-pir>;...

*Example:*
RtBrick-QoS-Policer = level=4,cir=1m,cbs=256;cir=2m,cbs=512,level=3

This attribute is supported by RADIUS access-accept and CoA request.

Updating a single policer level via CoA does not require to include other levels meaning that only the levels included in the attribute will be updated and all other levels remain unchanged. It is only possible to update existing policer levels dynamically but it is not possible to create a new level via RADIUS.

Assume the following example which adapts the just one level as well as all levels of a subscriber.

```
supervisor@rtbrick: op> show config forwarding-options class-of-service policer
policer-residential
{
  "rtbrick-config:policer": {
    "policer-name": "policer-residential",
    "level1-rates": {
      "cir": 1000,
      "cbs": 100,
      "pir": 1000,
      "pbs": 100
    },
    "level2-rates": {
      "cir": 4000,
      "cbs": 400,
      "pir": 4000,
      "pbs": 400
    },
    "level3-rates": {
      "cir": 0,
      "cbs": 800,
      "pir": 0,
      "pbs": 800
    },
    "level4-rates": {
      "cir": 0,
      "cbs": 800,
      "pir": 0,
      "pbs": 800
```

```
    },
    "levels": 4,
    "type": "two-rate-three-color"
  }
}
```

**RADIUS VSA Change Level 2 Only**

```
RtBrick-QoS-Policer: 2,12000,1200
```

**RADIUS VSA Change all Levels**

```
RtBrick-QoS-Policer: 1,2000,200;2,8000,800;3,0,800;4,0,800
```

All active dynamic policer level settings are stored in the table **global.access.1.subscriber.qos** to handover those information to forwarding infrastructure. This table can be also used to verify the dynamic policer settings for a given subscriber.

The CLI command `show subscriber <id> qos` displays those information in a nice human readable format.

```
supervisor@rtbrick: op> show subscriber 72339069014638610 qos
Subscriber-Id: 72339069014638610
    Interface: ifp-0/0/1
    Outer VLAN: 128
    Inner VLAN: 7
    IFL: ppp-0/0/1/72339069014638610
    Profile: qos_profile
    Parent: pon1
    Dynamic Ingress Policer Level 1:
        CIR: 2000 kbps CBS: 200 kbps
        PIR: 2000 kbps PBS: 200 kbps
    Dynamic Ingress Policer Level 2:
        CIR: 8000 kbps CBS: 800 kbps
        PIR: 8000 kbps PBS: 800 kbps
    Dynamic Ingress Policer Level 3:
        CIR: 0 kbps CBS: 800 kbps
        PIR: 0 kbps PBS: 800 kbps
    Dynamic Ingress Policer Level 4:
        CIR: 0 kbps CBS: 800 kbps
        PIR: 0 kbps PBS: 800 kbps
    Dynamic Shaper: shaper_voice
        Rate Low: 0 kbps
        Rate High: 2000 kbps
    Dynamic Shaper: shaper_session
        Rate Low: 2000 kbps
        Rate High: 14000 kbps
```

#### RADIUS QoS MFC

### *VSA 26-50058-66 - RtBrick-QoS-MFC*

The multifield classifier can be either derived from qos-profile or directly using this attribute which has priority of the qos-profile.

```
RtBrick-QoS-MFC = <mfc-name>
```

This attribute is supported by RADIUS access-accept and CoA request.

The subscriber management infrastructure does not check if a referenced multifield classifier exists or not. This is handled by forwarding infrastructure which continues processing the subscriber QoS settings as soon as the classifier was added.

The string received in these attributes should be stored as mf_classifier_name in the RADIUS attribute object and as well as in the corresponding subscriber-qos object.

#### RADIUS QoS Queues

### *VSA 26-50058-67 - RtBrick-QoS-Queues*

Subscribers can be associated with a QoS profile which is assigned via service-profile or directly via corresponding VSA (RtBrick-QoS-Profile). This profile is used to instantiate the QoS resources for the subscriber including schedulers, queues and shapers. The assigned queue instances can be updated using the RtBrick-QoS-Queues VSA (attribute 26-50058-67) which will apply to the QoS instance of the corresponding subscriber only, but not to the other subscribers using the same QoS profile. It is only possible to update existing queues dynamically but it is not possible to add queues via RADIUS.

The RtBrick-QoS-Queues value is a string which contains a list of multiple queue settings separated by semicolon. Each queue setting contains a queue name and queue size in bytes separated by comma.

```
RtBrick-QoS-Queues = <queue-name>,<size-bytes>;<queue-name>,<size-bytes>;...
```

This attribute can be also used as a key-value list which is automatically recognized by RBFS.

```
RtBrick-QoS-Queues = name=<queue-name>,size=<size-bytes>;name=...
```

This attribute is supported by RADIUS access-accept and CoA request.

Updating a single queue (e.g. best-effort) via CoA does not require to include other queues meaning that only the queues included in the attribute will be updated and all other queues remain unchanged.

The subscriber management infrastructure does not check if a referenced queue exists or not. This is handled by forwarding infrastructure which continues processing the subscriber QoS settings as soon as the queue was added.

All active dynamic queues are stored in the table **global.access.1.subscriber.qos.queue** to

handover those information to forwarding infrastructure. This table can be also used to verify the dynamic queues which are active for a given subscriber.

The CLI command `show subscriber <id> qos` displays those information in a nice human readable format.

```
supervisor@rtbrick: op> show subscriber 72339069014638610 qos
Subscriber-Id: 72339069014638610
    Interface: ifp-0/0/1
    Outer VLAN: 128
    Inner VLAN: 7
    IFL: ppp-0/0/1/72339069014638610
    Profile: qos_profile
    Parent: pon1
    Dynamic Ingress Policer Level 1:
        CIR: 2000 kbps CBS: 200 kbps
        PIR: 2000 kbps PBS: 200 kbps
    Dynamic Ingress Policer Level 2:
        CIR: 8000 kbps CBS: 800 kbps
        PIR: 8000 kbps PBS: 800 kbps
    Dynamic Ingress Policer Level 3:
        CIR: 0 kbps CBS: 800 kbps
        PIR: 0 kbps PBS: 800 kbps
    Dynamic Ingress Policer Level 4:
        CIR: 0 kbps CBS: 800 kbps
        PIR: 0 kbps PBS: 800 kbps
    Dynamic Shaper: shaper_voice
        Rate Low: 0 kbps
        Rate High: 2000 kbps
    Dynamic Shaper: shaper_session
        Rate Low: 2000 kbps
        Rate High: 14000 kbps
    Dynamic Queue: voice
        Size: 200000 byte
```

**RADIUS IGMP Attributes**

IGMP service can be dynamically enabled on a subscriber using Radius IGMP Attributes. These attributes are supported by RADIUS access-accept and CoA requests.

Following IGMP related attributes are supported:

***VSA 26-50058-71 - RtBrick-IGMP-Status***

This attribute can dynamically enable/disable IGMP for a subscriber

| Value | Code | Description |
|---|---|---|
| DISABLED | 0 | Disable IGMP |
| ENABLED | 1 | Enable IGMP |

*VSA 26-50058-72 - RtBrick-IGMP-Profile*

| |
|---|
| RtBrick-IGMP-Profile = <igmp-profile> |

This attribute specifies the IGMP-profile to be associated with the subscriber. IGMP profile is configured locally in IGMP with all the IGMP related attributes. This attribute can dynamically associate such a locally configured profile to a subscriber

The subscriber management infrastructure does not check if a referenced IGMP profile exists or not. This is handled by IGMP infrastructure which continues processing the subscriber IGMP settings as soon as the profile was added.

*VSA 26-50058-73 - RtBrick-IGMP-Version*

This attribute can specify the version of IGMP for a subscriber.

| Value | Code | Description |
|---|---|---|
| V1 | 1 | IGMP Version 1 (not supported) |
| V2 | 2 | IGMP Version 2 |
| V3 | 3 | IGMP Version 3 (Default version if this attribute is not set) |

*VSA 26-50058-74 - RtBrick-IGMP-Max-Members*

This integer attribute can specify the number of parallel streaming (maximum IGMP membership) for a subscriber.

The CLI command `show subscriber <id> igmp` displays those information.

```
supervisor@rtbrick: op> show subscriber 72339069014638706 igmp
Subscriber-Id: 72339069014638706
    Interface: ifp-0/0/1
    Outer VLAN: 128
    Inner VLAN: 7
    IFL: ppp-0/0/3/72339069014638706
    State: active
    Version: IGMPv3
    Profile: iptv-basic
    Max Members: 6
```

The states are `active`, `inactive` and `disabled`. The state is `inactive` if IGMP is enabled but the IPv4 protocol state is not active.

**RADIUS Connection Status Message Attribute**

*VSA 26-50058-139 - RtBrick-Connection-Status-Message*

This attribute allows to send a connection status message string via PPP LCP vendor extension

(RFC2513) to the client.

The connection status message is typically used to inform clients about the available bandwidth via RADIUS access-accept or CoA request.

```
RtBrick-Connection-Status-Message = <string>
```

Adding or changing the connection status message triggers to send a LCP vendor specific packet with OUI set to `f4:1e:5e` (RtBrick Inc.) and kind `0x01`.

*RFC 2513 - Vendor Specific Packet*

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Magic-Number                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      OUI                      |     Kind      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Value(s) ...
+-+-+-+-+-+-+-+-+
```

The value is encoded as TLV with type set to `0x01` and length including type and length field.

The actual status message string is limited to 247 bytes.

*Connection-Status-Message Option*

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     | Connection-Status-Message     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| ...
+-+-+
```

PPP clients with support for connection status message will respond with a vendor specific packet of kind `0x02` and the same OUI but no value to acknowledge the packet. All other clients either ignore or reject the message using PPP LCP code reject.

The connection status message will be retried 10 times with an interval of 3 seconds until the client has acknowledged or rejected the packet.

The status of the connection status message can be verified with the following command.

```
supervisor@leaf1: op> show pppoe session 72339069014638706 detail
Subscriber-Id: 72339069014638706
    State: ESTABLISHED
```

```
    Uptime: Mon Sep 13 09:29:42 GMT +0000 2021 (0:14:57.625027)
    Interface: ifp-0/0/1
    Outer VLAN: 128
    Inner VLAN: 7
...
    PPP LCP:
        State: OPENED
        Negotiated Protocols: PAP, IPCP, IP6CP
        Negotiated Parameters: MRU, AUTH, MAGIC
        Magic Number: 130827225 Peer: 2835676915
        MRU: 1492 Peer: 1492
        Echo Interval: 30 seconds
    PPP LCP Connection Status Message:
        State: ACCEPTED
        Message: SRU=10000#SRD=100000#
...
```

**RADIUS Ascend-Data-Filter Attribute**

***VSA 26-529-242 - Ascend-Data-Filter***

The Ascend-Data-Filter attribute describes per subscriber filter terms in a simple binary format as described in the following table. Multiple of those attributes in a single RADIUS access-accept or CoA request message are combined to a dynamic filter where each attribute itself is one filter term. The order of those attributes in the RADIUS message is used to order the corresponding terms in the filter. This means that the first filter in the RADIUS packet has priority over the next and or last filter in the RADIUS packet.

Changes in such a filter via CoA requires that all attributes of the new filter must be sent. Therefore adding a single filter term requires sending the existing filter terms plus the new one or general speaking sending the intended target filter.

This filter is installed as a global packet filter placed before policer in ingress and before scheduler in egress to prevent that traffic dropped here is counted in accounting or consuming rate credits.

| Field | Bytes | Values | Comments |
|---|---|---|---|
| Type | 1 | 0 = ignore<br><br>1 = IPv4<br><br>3 = IPv6 | |
| Action | 1 | 0 = discard<br><br>1 = accept<br><br>0x20 = redirect | |
| Direction | 1 | 0 = egress<br><br>1 = ingress | |

| Field | Bytes | Values | Comments |
|---|---|---|---|
| Reserved | 1 | 0 | |
| Source<br><br>Address | IPv4 = 4<br><br>IPv6 = 16 | source address | Match on source address is not supported for filters in ingress direction and automatically replaced with the subscriber addresses. |
| Destination<br><br>Address | IPv4 = 4<br><br>IPv6 = 16 | destination address | Match on destination address is not supported for filters in egress direction and automatically replaced with the subscriber addresses. |
| Source Prefix Length | 1 | 0 = ignore<br><br>prefix length | The number of leading one bits in the source address mask. Specifies the bits of interest. |
| Destination<br><br>Prefix<br><br>Length | 1 | 0 = ignore<br><br>prefix length | The number of leading one bits in the destination address mask. Specifies the bits of interest. |
| Protocol | 1 | 0 = ignore<br><br>IPv4 protocol number<br><br>IPv6 next header | |
| Reserved | 1 | 0 | |
| Source<br><br>Port | 2 | port number | UDP/TCP source port in network byte order (big endian) |
| Destination<br><br>Port | 2 | port number | UDP/TCP destination port in network byte order (big endian) |

| Field | Bytes | Values | Comments |
|---|---|---|---|
| Source Port Qualifier | 1 | 0 = no compare<br><br>1 = less than<br><br>2 = equal to<br><br>3 = greater than<br><br>4 = not equal to | The options 1, 3 and 4 are not implemented and mapped to 0 if received. |
| Destination Port Qualifier | 1 | 0 = no compare<br><br>1 = less than<br><br>2 = equal to<br><br>3 = greater than<br><br>4 = not equal to | The options 1, 3 and 4 are not implemented and mapped to 0 if received. |
| Not Used | 0 - N | 0 | Trailing bytes after destination port qualifier ignored. |

The filter can be deleted dynamically by sending a single Ascend-Data-Filter attribute with a zero value.

Example:

```
Ascend-Data-Filter = 0x01000100000000000a0afffe002000000000000000000000000000000000

    01          IPv4            // type
    00          discard         // action
    01          ingress         // direction
    00          -               // reserved
    00000000    -               // source address
    0a0afffe    198.51.100.101  // destination address
    00          -               // source prefix length
    20          /32             // destination prefix length
    00          -               // protocol
    00          -               // reserved
    0000        -               // source port
    0000        -               // destination port
    00          -               // source port qualifier
    00          -               // destination port qualifier
    00000...    -               // ignored
```

For ingress filters, it is not permitted to match based on the source address which is automatically replaced with the subscriber addresses. A similar limitation is

valid for egress filters matching on destination address which is also replaced with subscriber addresses.

A new proprietary filter action `0x20 (32)` is added for the redirect service. This can be used to enable HTTP Redirect for a subscriber using the ADF.

All active subscriber filters are stored in the table **global.access.1.subscriber.filter** to hand over that information to the forwarding infrastructure. This table can be also used to verify the filters which are active for a given subscriber.

The CLI command `show subscriber <id> acl` displays those filters in a nice human-readable format.

**RADIUS Access Line Attributes**

The access line identification and characterization information are defined in the Broadband Forum (BBF) formerly known DSL Forum attributes including Agent-Remote-Id and Agent-Circuit-Id.

See the following references for more information about access line attributes.

- RFC 4679 DSL Forum Vendor-Specific RADIUS Attributes
- RFC 6320 ANCP
- Broadband Forum TR-101
- draft-lihawi-ancp-protocol-access-extension-04

Those attributes will be sent in RADIUS access and accounting requests to the RADIUS server if learned from the underlying access protocol like PPPoE (BBF TR-101).

RBFS also provides the possibility to set those values via RADIUS by using the same attributes in the RADIUS access-accept or CoA request message.

*Access Line Attributes Supported (RADIUS, PPPoE and L2TP):*

| Attribute | Name |
|---|---|
| 26-3561-1 | Agent-Circuit-Id |
| 26-3561-2 | Agent-Remote-Id |
| 26-3561-18 | PON-Access-Line-Attributes |
| 26-3561-129 | Actual-Data-Rate-Upstream |
| 26-3561-130 | Actual-Data-Rate-Downstream |
| 26-3561-131 | Minimum-Data-Rate-Upstream |
| 26-3561-132 | Minimum-Data-Rate-Downstream |
| 26-3561-133 | Attainable-Data-Rate-Upstream |
| 26-3561-134 | Attainable-Data-Rate-Downstream |
| 26-3561-135 | Maximum-Data-Rate-Upstream |

| Attribute | Name |
|---|---|
| 26-3561-136 | Maximum-Data-Rate-Downstream |
| 26-3561-137 | Minimum-Data-Rate-Upstream-Low-Power |
| 26-3561-138 | Minimum-Data-Rate-Downstream-Low-Power |
| 26-3561-139 | Maximum-Interleaving-Delay-Upstream |
| 26-3561-140 | Actual-Interleaving-Delay-Upstream |
| 26-3561-141 | Maximum-Interleaving-Delay-Downstream |
| 26-3561-142 | Actual-Interleaving-Delay-Downstream |
| 26-3561-144 | Access-Loop-Encapsulation |
| 26-3561-145 | DSL-Type |
| 26-3561-151 | PON-Access-Type |
| 26-3561-155 | Expected-Throughput-Upstream |
| 26-3561-156 | Expected-Throughput-Downstream |
| 26-3561-157 | Attainable-Expected-Throughput-Upstream |
| 26-3561-158 | Attainable-Expected-Throughput-Downstream |
| 26-3561-159 | Gamma-Data-Rate-Upstream |
| 26-3561-160 | Gamma-Data-Rate-Downstream |
| 26-3561-161 | Attainable-Gamma-Data-Rate-Upstream |
| 26-3561-162 | Attainable-Gamma-Data-Rate-Downstream |
| 26-3561-176 | ONT-ONU-Average-Data-Rate-Downstream |
| 26-3561-177 | ONT-ONU-Peak-Data-Rate-Downstream |
| 26-3561-178 | ONT-ONU-Maximum-Data-Rate-Upstream |
| 26-3561-179 | ONT-ONU-Assured-Data-Rate-Upstream |
| 26-3561-180 | PON-Tree-Maximum-Data-Rate-Upstream |
| 26-3561-181 | PON-Tree-Maximum-Data-Rate-Downstream |

This table includes already the new attributes defined in draft-lihawi-ancp-protocol-access-extension-04 which results in possible changes in case the draft is updated.

Changes in at least one of those line attributes via CoA trigger a RADIUS interim accounting request with the new values.

Changes in actual data rate upstream/downstream via CoA request may trigger the L2TP LAC to send a CSUN request with updated connection speed to the LNS depending on the actual L2TP configuration.

**RADIUS L2TP**

Tunneling of PPPoE sessions via L2TPv2 (L2TP LAC) is supported on RBFS and can be controlled via

RADIUS as described in RFC 2868 RADIUS Attributes for Tunnel Protocol Support with some limitations:

- No support of FQDN format for IP addresses
- No support Tunnel-Medium-Type other than IPv4

To establish a PPPoE session via L2TP, the tunnel-type must be configured as L2TP. This configuration can be achieved either for local users or by utilizing the corresponding tunnel-type attribute through RADIUS.

Defining an L2TP configuration profile is essential, which can be referenced through an access configuration profile or by employing the appropriate RADIUS VSA. The actual tunnels may either be defined locally via an L2TP pool configuration or set up dynamically through RADIUS.

```
# FreeRADIUS
# RADIUS authenticated with local defined tunnels.
"radius@l2tp" Cleartext-Password := "test"
    Service-Type = Framed-User,
    Framed-Protocol = PPP,
    Tunnel-Type:0 = L2TP

# RADIUS authenticated with dynamic tunnels.
"tunnel@rl2tp" Cleartext-Password := "test"
    Service-Type = Framed-User,
    Framed-Protocol = PPP,
    Tunnel-Type:1 = L2TP,
    Tunnel-Client-Endpoint:1 = "192.0.2.1",
    Tunnel-Server-Endpoint:1 = "192.0.2.2",
    Tunnel-Client-Auth-Id:1 = "bng",
    Tunnel-Server-Auth-Id:1 = "lns",
    Tunnel-Preference:1 = 100,
    Tunnel-Password:1 = "test",
    Tunnel-Type:2 += L2TP,
    Tunnel-Client-Endpoint:2 += "192.0.2.3",
    Tunnel-Server-Endpoint:2 += "192.0.2.4",
    Tunnel-Client-Auth-Id:2 += "BNG",
    Tunnel-Server-Auth-Id:2 += "LNS2",
    Tunnel-Preference:2 += 200,
    Tunnel-Password:2 += "test"
```

In addition to the standard attributes, the following vendor-specific attributes are supported for L2TP.

RFC and draft compliance are partial except as specified.

***VSA 26-50058-40 - RtBrick-L2TP-Pool***

Instead of RADIUS tunnel attributes it is also possible to configure local L2TP tunnel pools and assign them with this attribute.

***VSA 26-50058-41 - RtBrick-L2TP-Profile***

The default L2TP configuration profile assigned in the access configuration profile can be changed in RADIUS access-accept to allow different L2TP configurations based on tunnel endpoints. This is typically used to enable or disable L2TP CSUN updates (RFC5515).

***VSA 26-50058-42 - RtBrick-L2TP-Tx-Connect-Speed***

***VSA 26-50058-43 - RtBrick-L2TP-Rx-Connect-Speed***

These attributes are supported in RADIUS access-accept and CoA requests to set the corresponding L2TP to connect speed values directly. Per default, the L2TP connect speed is derived from the actual data rate upstream/downstream of the configured source (PPPoE-IA or RADIUS) if present or set to port speed which can be overridden with the connect speed attributes. Changes in connect speed via CoA request will trigger the L2TP LAC to send a CSUN request to the LNS if enabled.

These two attributes are defined as 4-byte integers with speed defined in bits per second (bps).

**RADIUS Lawful Interception**

Lawful interception (LI) refers to the facilities in telecommunications and telephone networks that allow law enforcement agencies (LEA) with court orders or other legal authorization to selectively intercept individual subscribers. The term mediation device (MED) used in this document describes the element which receives and optionally converts the intercepted traffic into the format expected by the law enforcement agencies of the corresponding countries.

All of the following attributes must be present in RADIUS access-accept or CoA request to control lawful interception (LI) via RADIUS. Those attributes are salt encrypted using the algorithm described in RFC 2868 for the Tunnel Password. This encryption algorithm is defined for RADIUS access-accept messages only. To support CoA requests the request authenticator should be replaced with 16 zero bytes which is common industry standard.

The LI action NOOP can be used to obfuscate lawful interception requests (fake requests) to prevent just the presence of those attributes indicating that a subscriber is intercepted. LI requests via RADIUS will show up in the same table as requests via REST or HTTP RPC API (global.access.1.li_request).

> Failed LI activations are not signaled via RADIUS to prevent that just the presence of CoA response NAK shows that LI request is not fake (action NOOP).

***VSA 26-50058-140 - RtBrick-LI-Action (salt encrypted integer)***

| Value | Code | Description |
|---|---|---|
| NOOP | 0 | No action / Ignore LI request |
| ON | 1 | Start LI / Add LI request |
| OFF | 2 | Stop LI / Delete LI request |

***VSA 26-50058-141 - RtBrick-LI-Identifier (salt encrypted integer)***

Device unique lawful interception identifier (LIID) within the range from 1 to 4194303.

***VSA 26-50058-142 - RtBrick-LI-Direction (salt encrypted integer)***

| Value | Code | Description |
|---|---|---|
| INGRESS | 1 | Ingress mirroring only (from subscriber) |
| EGRESS | 2 | Egress mirroring only (to subscriber) |
| BOTH | 3 | Bidirectional mirroring (from and to subscriber) |

***VSA 26-50058-143 - RtBrick-LI-MED-Instance (salt encrypted string)***

Routing instance through which the mediation device is reachable.

***VSA 26-50058-144 - RtBrick-LI-MED-IP (salt encrypted IPv4 address)***

IPv4 address of the mediation device.

***VSA 26-50058-145 - RtBrick-LI-MED-Port (salt encrypted integer)***

UDP port between 49152 and 65535 are set in the mirrored traffic.

**RADIUS Framed Routes**

The framed-route attributes are used by the RADIUS to install specific routes for a given subscriber. The corresponding next-hop information is automatically derived from the subscriber context. This functionality is typically used to route additional customer networks through the client device (CPE).

Those attributes are supported via RADIUS access-accept only. Each of those attributes may occur multiple times to install various routes. The actual amount of routes supported is limited by the maximum RADIUS packet size only.

The framed-route attributes are defined in RFC 2865 for IPv4 and RFC 3162 for IPv6 but the actual content of those attributes is implementation dependent. RBFS is expecting a string with a prefix followed by prefix length. The prefix can be followed by further routing options like costs, metric or even next-hop which are currently ignored.

That there is no support for multiple subscriber sessions having same framed route, neither in same or in different routing-instances.

***Attribute 22 - Framed-Route***

```
Framed-Route = "198.51.100.69/24"
```

***Attribute 99 - Framed-IPv6-Route***

```
Framed-IPv6-Route = "2001:db8:0:90::/32"
```

**RADIUS Subscriber ACLs (Filters)**

The subscriber ACL attributes are used by the RADIUS server to install a specific ACL, which has already been configured in the BNG, for a subscriber. For information about Subscriber ACL Configuration see 'Subscriber ACLs and HTTP Redirect Service' guide. After the Subscriber ACLs are configured with the name and the rules, use the following RADIUS attributes dynamically for adding or changing those ACLs from RADIUS. These attributes are supported in `access-accept` and `CoA`.

The subscriber ACL attributes are used by the RADIUS server to install a specific ACL, which has already been configured in the BNG, for a subscriber. For information about Subscriber ACL Configuration, see the 'Subscriber ACLs and HTTP Redirect Service' guide. After the Subscriber ACLs are configured with the name and the rules, use the following RADIUS attributes to add or change those ACLs from RADIUS. These attributes are supported in `access-accept` and `CoA`.

When a CoA is issued with one of those ACL attributes or a service profile, it will always reset all existing filters, even if the new service profile does not define any filters. If there are filters specified within the service profile along with subscriber ACL attributes, these subscriber ACL attributes take precedence. The filters are merged by first applying all the filters from the service profile and then adding or replacing filters using the attributes defined here. The final resulting filter set is then applied to the subscriber.

Filters that were applied first but are not defined in the new set will be deleted. Consequently, adding a single filter via CoA will remove all other existing filters. This approach is designed to provide a more declarative way of defining the desired target state without depending on the existing state.

- ***VSA 26-50058-76 - RtBrick-IPv4-ACL-In***

  ```
  Attach IPv4 subscriber ingress ACL (ACL in upstream direction)
  ```

- ***VSA 26-50058-77 - RtBrick-IPv4-ACL-Out***

  ```
  Attach IPv4 subscriber ingress ACL (ACL in downstream direction)
  ```

- ***VSA 26-50058-78 - RtBrick-IPv6-ACL-In***

  ```
  Attach IPv6 subscriber ingress ACL (ACL in upstream direction)
  ```

- ***VSA 26-50058-79 - RtBrick-IPv6-ACL-Out***

  ```
  Attach IPv6 subscriber egress ACL (ACL in downstream direction)
  ```

Example:

RtBrick-IPv4-ACL-In = “ipoe-sub1-http-acl“

To detach any of the preceding ACLs and to delete the ACL associated with a subscriber, `null` string can be used with the corresponding attribute.

Example:

RtBrick-IPv4-ACL-In = ““

**RADIUS HTTP Redirect URL**

When HTTP Redirect is enabled for a subscriber either through Ascend-Data-Filter or through the subscriber ACLs, HTTP packets from the subscriber get redirected to the URL set by this attribute:

***VSA 26-50058-75 - RtBrick-HTTP-Redirect-URL***

This attribute is supported on RADIUS `access-accept` and `CoA` requests. This attribute allows the overwriting of the URL.

For example:

RtBrick-HTTP-Redirect-URL = “http://portal.rtbrick.com”

To reset the URL:

RtBrick-HTTP-Redirect-URL = “”

**RADIUS Sourced DHCP Options**

RBFS allows for centralized configuration of DHCP options directly on a RADIUS server, enabling dynamic distribution of these options on a per-subscriber basis. This approach, known as **RADIUS Sourced DHCP Options**, shifts the responsibility of defining and managing DHCP options to the RADIUS server.

Since RBFS performs minimal processing and validation on these options, they are treated as **raw data**, meaning they are forwarded without modification or extensive error checking.

- ***VSA 26-50058-30 - RtBrick-DHCPv4-Client-Options***
- ***VSA 26-50058-32 - RtBrick-DHCPv6-Client-Options***

The client-option attributes allow DHCP options to be included as part of the Access-Accept message. These options are then inserted into outgoing packets directed toward the subscribers DHCP client, regardless of whether RBFS is operating in IPoE relay/proxy mode or server mode.

- ***VSA 26-50058-31 - RtBrick-DHCPv4-Server-Options***
- ***VSA 26-50058-33 - RtBrick-DHCPv6-Server-Options***

Unlike client-option attributes, the server-option attributes are only supported in IPoE relay/proxy

mode and are specifically designed to modify packets being forwarded to a DHCP server. This functionality provides additional configuration flexibility when RBFS operates as a relay or proxy between the client and the DHCP server.

For example, server-option attributes can be used to insert specific DHCP options into requests sent to the DHCP server, enabling it to select the appropriate IP address pool based on the provided parameters.

If a RADIUS server includes multiple instances of the same DHCP option within a single Access-Accept message, RBFS will concatenate them into a single data stream. However, the order of the attributes in the packet is crucial for proper reassembly.

For successful concatenation, the option instances must appear consecutively in the RADIUS response. If other attributes are interleaved between multiple instances of **RtBrick-DHCP-Client-Options**, those can't be concatenated.

***FreeRADIUS Example:***

```
"02:00:00:00:00:01@ipoe" Cleartext-Password := "ipoe"
    Service-Type = Framed-User,
    Class = IPOE,
    RtBrick-DHCPv4-Client-Options = "0x39020240",
    RtBrick-DHCPv4-Client-Options += "0x3d0701001094000008",
    RtBrick-DHCPv4-Server-Options = "0x39020240",
    RtBrick-DHCPv6-Server-Options = "0x001100100000c38a000200084353522D56365044",
    RtBrick-DHCPv6-Client-Options = "0x001100100000c38a000100084353522D56365044"
```

***Example Option:***

```
0x39020240

  39    Code      // DHCP Max Msg Size
  02    Length    // 2 Byte
  0240  Data      // 576 Byte
```

Since options are added as raw data, they must be in network byte order.

DHCPv6 and DHCPv4 operate differently when it comes to the inclusion of client and server options within DHCP messages.

In DHCPv6, server options are incorporated into the outermost DHCPv6 relay header. This specific header is added by the relay agent (RBFS). On the other hand, client options are added to the innermost DHCPv6 message, but this only occurs if the message type is either an advertisement (type 2) or a reply (type 7).

Conversely, in DHCPv4, the placement of client and server options is based on specific message types. Client options are exclusively added to offer (type 2) and ack (type 5) messages. In contrast, server options are only included in discovery (type 1) and request (type 3) messages.

The following DHCP options are not permitted and will be filtered out if attempted to be added via RADIUS:

**DHCPv4:**

- **0:** Pad Option
- **51:** IP Address Lease Time
- **52:** Option Overload
- **53:** DHCP Message Type
- **54:** Server Identifier
- **55:** Parameter Request List
- **255:** End

**DHCPv6:**

- **1:** OPTION_CLIENTID
- **2:** OPTION_SERVERID
- **3:** OPTION_IA_NA
- **6:** OPTION_ORO
- **8:** OPTION_ELAPSED_TIME
- **25:** OPTION_IA_PD

The DHCP options will add received options even if they are already present in the original packet unless they are filtered out as not allowed or invalid. For the DHCPv4 server options listed below, the original option is removed before the same option is added via the DHCPv4 server options to prevent duplicate options. Other options may still become duplicates.

- Client-Identifier (61)
- Relay-Agent (82)

**RADIUS Terminate Codes**

***VSA 26-50058-27 - RtBrick-Terminate-Code***

The RtBrick-Terminate-Code is sent along with the standard Acct-Terminate-Cause (attribute 49). The list below shows the RtBrick termination codes with the mapping to the standard cause.

| RtBrick-Terminate-Code | Acct-Terminate-Cause (RADIUS Attribute 49) | Description |
|---|---|---|
| 0 | 10 (NAS Request) | NA |
| 1 | 10 (NAS Request) | Subscriber Management Unknown Error |
| 2 | 9 (NAS Error) | SubscriberD Internal Error |

| RtBrick-Terminate-Code | Acct-Terminate-Cause (RADIUS Attribute 49) | Description |
|---|---|---|
| 3 | 9 (NAS Error) | PPPoED Internal Error |
| 4 | 9 (NAS Error) | PPPoED Object Deleted<br>*This code is used if an PPPoE session object is deleted which is an indication for a crash or some other issues in PPPoE daemon.* |
| 5 | 9 (NAS Error) | L2TPD Internal Error |
| 6 | 9 (NAS Error) | L2TPD Object Deleted<br>*This code is used if an L2TP session object is deleted which is an indication for a crash or some other issues in L2TP daemon.* |
| 7 | 9 (NAS Error) | AAA Profile Not Found |
| 8 | 9 (NAS Error) | RADIUS Profile Not Found |
| 9 | 9 (NAS Error) | Authentication Type Missing |
| 10 | 9 (NAS Error) | Authentication Order Missing or Invalid |
| 11 | 10 (NAS Request) | Authentication Failure |
| 12 | 10 (NAS Request) | Local Authentication Failed |
| 13 | 10 (NAS Request) | Accounting-Request-On Wait |
| 14 | 9 (NAS Error) | No RADIUS Authentication Server Configured |
| 15 | 10 (NAS Request) | RADIUS Authentication Failed |
| 16 | 17 (User Error) | Authentication Rejected |
| 17 | 17 (User Error) | Local Authentication Rejected |
| 18 | 17 (User Error) | RADIUS Authentication Rejected |
| 19 | 5 (Session Timeout) | Session Timeout |
| 20 | 4 (Idle Timeout) | Idle Timeout |
| 21 | 6 (Admin Reset) | Clear Session |
| 22 | 6 (Admin Reset) | RADIUS CoA Disconnect |
| 23 | 9 (NAS Error) | PPPoE Unknown Error |
| 24 | 1 (User Request) | PPPoE PADT Received |
| 25 | 9 (NAS Error) | PPPoE LCP Error |
| 26 | 10 (NAS Request) | PPPoE LCP Generic Error |
| 27 | 1 (User Request) | PPPoE LCP Terminate Request Received |

| RtBrick-Terminate-Code | Acct-Terminate-Cause (RADIUS Attribute 49) | Description |
|---|---|---|
| 28 | 10 (NAS Request) | PPPoE LCP Maximum Reject / NAK Exceeded |
| 29 | 10 (NAS Request) | PPPoE LCP Negotiation Failed |
| 30 | 10 (NAS Request) | PPPoE LCP Configuration-Request Exceeded |
| 31 | 10 (NAS Request) | PPPoE LCP Echo-Request Timeout Exceeded |
| 32 | 9 (NAS Error) | PPPoE PAP Error |
| 33 | 9 (NAS Error) | PPPoE CHAP Error |
| 34 | 9 (NAS Error) | PPPoE IPCP Error |
| 35 | 9 (NAS Error) | PPPoE IP6CP Error |
| 36 | 9 (NAS Error) | PPPoE NCP Initialization Failed |
| 37 | 10 (NAS Request) | PPPoE NCP Down |
| 38 | 6 (Admin Reset) | PPPoE Clear Session |
| 39 | 10 (NAS Request) | PPPoE Upper Layer Down |
| 40 | 8 (Port Error) | PPPoE Interface Down |
| 41 | 10 (NAS Request) | PPPoE Configuration Deleted |
| 42 | 9 (NAS Error) | PPPoE Access Configuration Profile Not Found |
| 43 | 9 (NAS Error) | L2TP Unknown Error |
| 44 | 10 (NAS Request) | L2TP Tunnel Down |
| 45 | 10 (NAS Request) | L2TP Tunnel Dead |
| 46 | 10 (NAS Request) | L2TP Tunnel Deleted |
| 47 | 10 (NAS Request) | L2TP No Tunnel Available |
| 48 | 10 (NAS Request) | L2TP CDN Request |
| 49 | 9 (NAS Error) | L2TP Profile Not Found |
| 50 | 9 (NAS Error) | L2TP Access Configuration Profile Not Found |
| 51 | 9 (NAS Error) | L2TP No Tunnel Pool Error |
| 52 | 9 (NAS Error) | L2TP Local Tunnel Pool Error |
| 53 | 9 (NAS Error) | L2TP RADIUS Tunnel Pool Error |
| 54 | 6 (Admin Reset) | L2TP Clear Session |
| 55 | 9 (NAS Error) | Access Configuration Profile Not Found |

| RtBrick-Terminate-Code | Acct-Terminate-Cause (RADIUS Attribute 49) | Description |
|---|---|---|
| 56 | 9 (NAS Error) | Local IPv4 Address Pool Not Found |
| 57 | 10 (NAS Request) | Local IPv4 Address Pool Exhausted |
| 58 | 9 (NAS Error) | Local IPv6 Prefix Pool Not Found |
| 59 | 10 (NAS Request) | Local IPv6 Prefix Pool Exhausted |
| 60 | 9 (NAS Error) | Local IPv6 Delegated Prefix Pool Not Found |
| 61 | 10 (NAS Request) | Local IPv6 Delegated Prefix Pool Exhausted |
| 62 | 10 (NAS Request) | PPPoE CHAP Response Timeout |
| 63 | 10 (NAS Request) | L2TP Session Deleted |
| 64 | 10 (NAS Request) | Duplicate IPv4 address detected |
| 65 | 10 (NAS Request) | Duplicate IPv6 prefix detected |
| 66 | 10 (NAS Request) | Duplicate delegated IPv6 prefix detected |
| 67 | 9 (NAS Error) | Routing instance not found |
| 68 | 6 (Admin Reset) | Clear Session Force |
| 69 | 6 (Admin Reset) | Test AAA Request Object Deleted |
| 70 | 9 (NAS Error) | IPoED Object Deleted<br>*This code is used if an IPoE subscriber object is deleted which is an indication for a crash or some other issues in IPoE daemon.* |
| 71 | 9 (NAS Error) | IPoED Internal Error |
| 72 | 9 (NAS Error) | IPoE Unknown Error |
| 73 | 10 (NAS Request) | IPoE Upper Layer Down |
| 74 | 10 (NAS Request) | IPoE Lower Layer Down |
| 75 | 10 (NAS Request) | IPoE Interface Down |
| 76 | 10 (NAS Request) | IPoE Configuration Deleted |
| 77 | 10 (NAS Request) | SubscriberD Request |
| 78 | 10 (NAS Request) | L2BSA Service Deleted |
| 79 | 10 (NAS Request) | L2BSA Service Added |
| 80 | 6 (Admin Reset) | IPoE Clear Subscriber |
| 81 | 6 (Admin Reset) | PPPoE Clear Session Force |
| 82 | 9 (NAS Error) | IPoE Invalid Gateway IFL |

| RtBrick-Terminate-Code | Acct-Terminate-Cause (RADIUS Attribute 49) | Description |
|---|---|---|
| 83 | 9 (NAS Error) | FWD Invalid |
| 84 | 9 (NAS Error) | FWD Failed |
| 85 | 9 (NAS Error) | FWD Deleted |
| 86 | 10 (NAS Request) | Instance Deleted |
| 87 | 10 (NAS Request) | Redundancy Switchover |
| 88 | 10 (NAS Request) | Redundancy Active Node Termination |
| 89 | 10 (NAS Request) | Redundancy Active Node Down |
| 90 | 9 (NAS Error) | Redundancy Invalid State |
| 91 | 9 (NAS Error) | Redundancy Invalid State for Sync |
| 92 | 9 (NAS Error) | Redundancy Validation Failed |
| 93 | 10 (NAS Request) | Redundancy Config Deleted |
| 94 | 9 (NAS Error) | Redundancy Stale Timer Expiry |
| 95 | 9 (NAS Error) | IPv4 In ACL Validation Failed |
| 96 | 9 (NAS Error) | IPv4 Out ACL Validation Failed |
| 97 | 9 (NAS Error) | IPv6 In ACL Validation Failed |
| 98 | 9 (NAS Error) | IPv6 Out ACL Validation Failed |
| 99 | 9 (NAS Error) | IP Address Allocation Failed |
| 100 | 9 (NAS Error) | No IP Address |
| 101 | 6 (Admin Reset) | Subscriber Deleted |
| 102 | 10 (NAS Request) | Redundancy No Interface To Assign *This code is used if a forwarding subscriber object is deleted which is an indication of a crash or some other issues in the forwarding infrastructure.* |
| 103 | 10 (NAS Request) | IPoE DHCPv4 Released |
| 104 | 10 (NAS Request) | IPoE DHCPv4 Expired |
| 105 | 10 (NAS Request) | IPoE DHCPv4 Failed |
| 106 | 10 (NAS Request) | IPoE DHCPv6 Released |
| 107 | 10 (NAS Request) | IPoE DHCPv6 Expired |
| 108 | 10 (NAS Request) | IPoE DHCPv6 Failed |
| 109 | 10 (NAS Request) | IPoE DHCPv4 Released DHCPv6 Released |

| RtBrick-Terminate-Code | Acct-Terminate-Cause (RADIUS Attribute 49) | Description |
|---|---|---|
| 110 | 10 (NAS Request) | IPoE DHCPv4 Released DHCPv6 Expired |
| 111 | 10 (NAS Request) | IPoE DHCPv4 Released DHCPv6 Failed |
| 112 | 10 (NAS Request) | IPoE DHCPv4 Expired DHCPv6 Released |
| 113 | 10 (NAS Request) | IPoE DHCPv4 Expired DHCPv6 Expired |
| 114 | 10 (NAS Request) | IPoE DHCPv4 Expired DHCPv6 Failed |
| 115 | 10 (NAS Request) | IPoE DHCPv4 Failed DHCPv6 Released |
| 116 | 10 (NAS Request) | IPoE DHCPv4 Failed DHCPv6 Expired |
| 117 | 10 (NAS Request) | IPoE DHCPv4 Failed DHCPv6 Failed |
| 118 | 10 (NAS Request) | Invalid RADIUS Authentication Response |
| 119 | 10 (NAS Request) | PPPoE and IPoE on same VLAN not Supported |

It is recommended to raise operational alarms for every termination cause received with a value of 9 (NAS Error) for further investigations.

**RADIUS CoA Error Handling**

CoA requests should be retried a few times in the unlikely event of CoA NAK with error-cause (RFC 5176 attribute 101) other than session-context-not-found (503) or missing-attribute (402). There is no rollback of failed CoA requests to the former state but retrying the same request is supported because changes are implemented in a more declarative way.

### 5.2.4. RADIUS Accounting

RADIUS accounting servers handle accounting records for subscribers. The subscriber daemon transmits RADIUS Accounting-Start, Interim and Stop messages to these servers. Accounting is the process of tracking subscriber activity and network resource usage in a subscriber session. This includes the session time called time accounting and the number of packets and bytes transmitted during the session called volume accounting.

A RADIUS Acct-Status-Type attribute is used by the RADIUS client (subscriber daemon) to mark the start of accounting (for example, upon booting) by specifying Accounting-On and to mark the end of accounting (for example, just before a scheduled reboot) by specifying Accounting-Off. This message is often used by RADIUS servers to automatically close/terminate all open accounting records/sessions for the corresponding client and therefore must not be sent to servers belonging to a profile which was already used/started for accounting.

Per default, the assumption is that all servers referenced by a RADIUS profile share the same states and therefore accounting-on must be only sent to one of those before the first accounting-start is

sent.

RADIUS Accounting-On/Off messages are disabled by default and can be optionally enabled in the RADIUS profile configuration. There is also an additional configuration option to optionally wait for Accounting-On response which prevents any new session until accounting has started meaning that Accounting-On response is received.

> Accounting-Off is currently not implemented!

RADIUS accounting requests are often used for billing and therefore should be able to store and retry over a longer period (common up to 24 hours or more) which can be optionally enabled in the RADIUS profile configuration using the accounting backup configuration option.

**Time Accounting**

All accounting relevant timestamps are retrieved using the UNIX system call clock_gettime internally stored in the datastore as 64 bit microseconds timestamp with the lower order 32 bit representing the seconds since January 1, 1970 00:00 UTC which is also known as epoch or unix timestamp and the higher order 32 bit representing the microseconds which are per definition less than 1.000.000 (one second). The seconds part is counted in full seconds which is similar to always rounded down.

The RADIUS attribute event-timestamp (type 55) defined in RFC2869 is included in each RADIUS Accounting-Request packet to record the time that this event occurred in seconds since January 1, 1970 00:00 UTC. A max allowed deviation of 500 milliseconds of a timestamp in seconds requires mathematical rounding of the internal 64 bit microseconds timestamps to the 32 bit RADIUS event-timestamp in seconds. This means to round down if the microseconds part of the timestamp is less than 0.5 seconds and rounded up if equal or greater than 0.5 seconds.

***Start Timestamps***

In RBFS the timestamp indicating the PPPoE session start will be generated after successful negotiation of at least one of the PPP network control protocols (IPCP for IPv4 or IP6CP for IPv6). In case of L2TP tunneled sessions (LAC), the successful L2TP session setup after sending ICCN is used as a start timestamp. In both cases this is also the trigger for the RADIUS accounting-request start where this timestamp is used as RADIUS event-timestamp.

***Stop Timestamps***

The timestamp indicating the session stop will be generated more or less immediately after termination request (e.g. timeout, user request, nas request, ...).

The RADIUS attribute Acct-Session-Time (type 46) defined in RFC2866 indicates how many seconds the user has received service and can only be present in Accounting-Request records where the Acct-Status-Type is set to Interim or Stop. This time will be calculated based on the internal session start and stop time in microseconds and mathematical rounded to seconds. For RADIUS accounting interim requests the current event time is used instead of stop time.

> RFC and draft compliance are partial except as specified.

**Volume Accounting**

Based on §45g of the German Telecommunications Act (TKG), the maximum deviation for accounting relevant counters must be less than 1% per billing period which is typically one month.

The whole RBFS architecture is based on an event-based publish and subscribe model using the BDS infrastructure. Each process publishes all information and states to BDS and interested processes subscribe to that information. The whole inter process communications (IPC) is replaced by BDS table operations which allows to build a hyper scaled distributed software system. This means related to accounting that each counter is requested from data plane and updated in BDS unsolicited based on configurable intervals typically set between 5 to 30 seconds. This applies to any type of counters like interface counters or QoS statistics. This means that CLI commands or API requests will return the last updated counter and not the counter of the time of request. Therefore each BDS object contains a timestamp indicating the time of the last counter update which allows to use these counters with the required accuracy.

Subscriber volume accounting is based on multiple sources like logical interface (LIF), class/queue-, policer- and control-traffic statistics.

All those counters are layer 2 (L2) counters and some of them may reset through configuration changes like QoS counters after applying a new QoS profile. Therefore the subscriber management application daemon is doing some post processing of all counters to ensure that no accounting information is lost and to calculate the final accounting values. This includes also optional configurable counter adjustments.

There is a BDS database object in the table **local.access.subscriber.accounting** created for each subscriber which stores all counters and attributes required to calculate the actual volume counters. This object is created together with the subscriber and automatically deleted if the actual subscriber object is removed.

The volume accounting counters must not reset if the actual hardware counter is deleted or has changed/reset to zero. Therefore instead of using the source counter directly, the delta since the last interval is calculated and added to the final counter.

The usage of callback functions to counter delete and change events ensures that no accounting relevant information will be lost through reconfiguration of subscriber sessions.

The function which removes the subscriber counters from the data plane will receive the final counters before removal and update them into BDS counter objects before this is deleted.

***Interims Volume Counters***

RADIUS accounting interim requests will also use the last updated counters with current time as event-timestamp.

***Stop Volume Counters***

The session termination might be triggered by REST API, CLI (clear subscriber …) RADIUS CoA disconnect request, session timeout, idle timeout or client request (PADT). The corresponding RADIUS accounting stop requests will be delayed to wait for the final counter update but uses the timestamp of the terminate request.

#### Interims Accounting

To receive RADIUS counters in fast intervals the corresponding session interim accounting interval can be set depending on the actual needs to any value between one second and multiple hours or days (recommended is at least 30 seconds). This interval can be also updated via CoA request using the Acct-Interim-Interval attribute which triggers an interim accounting immediately and uses the new interval from now onwards. This can be also used to request interim accounting requests on demand. Sending a CoA request with the applied interval triggers an accounting request but the original interval is not changed.

### 5.2.5. RADIUS Counters

The packets and byte counters of each session, traffic class (class 0 - 7) as well as policer counters (level 1 to 4) are supported via RADIUS accounting interims and stop messages. Those counters are layer two counters per default which can be adjusted using correction values and factors from the AAA configuration profile.

Subscriber accounting is based on multiple sources like LIF-, class/queue-, policer- and control-traffic statistics which are described below.

- The InLIF (Incoming Logical Interface) stats count all traffic received on the logical interface including control traffic are traffic dropped by ingress policer.
- The policer stats count all traffic accepted by policer (color green) per level (1-4). Ingress control traffic will be hit by a separate control plane policer and therefore not counted in the session policer stats.
- The class/queue stats count all egress traffic except control traffic which is directly sent to the IFP.
- The OutLIF (Outgoing Logical Interface) stats count all traffic sent on the logical interface except control traffic which is directly sent to the IFP.
- The control stats count all traffic received and sent from or to the control plane for a given subscriber (counted in PPPoE daemon for PPPoE sessions).

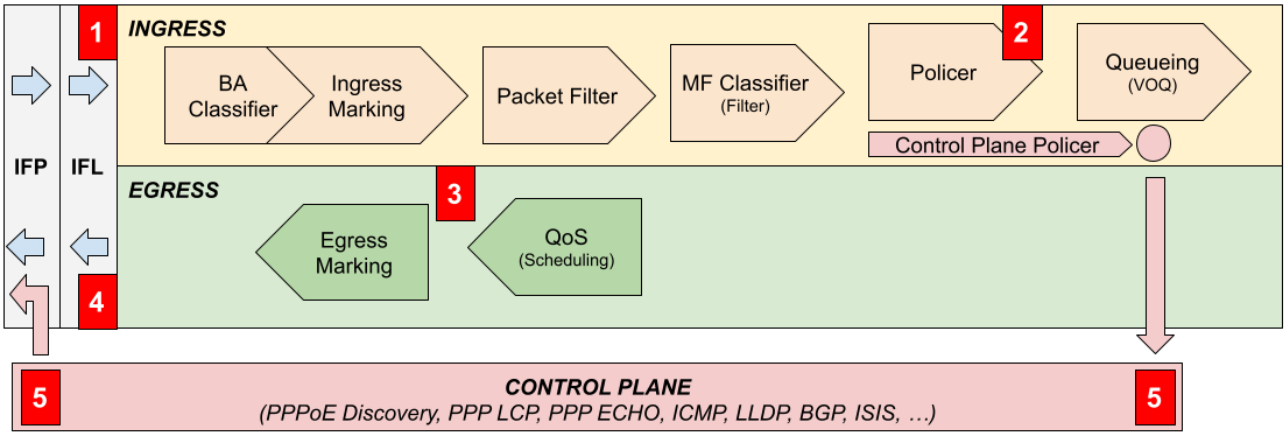



*Figure 82. RADIUS Counters*

Because counter updates are not atomic operations, the sum of all class counters

> might be different from the session counters.

The counted bytes per packet can be adjusted as described in chapter Configuring Accounting Adjustments of the Subscriber Management Configuration Guide.

All subscriber accounting attributes and counters are stored without adjustment (L2) in the table **local.access.subscriber.accounting** and remain until the subscriber session is finally removed after response to RADIUS accounting stop.

The command `show subscriber <id> accounting` displays the adjusted subscriber accounting information. It is also possible to display the values before adjustment using `show subscriber <id> accounting origin`.

```
supervisor@rtbrick: op> show subscriber 72339069014638637 accounting
Subscriber-Id: 72339069014638637
    IFL: ppp-0/0/1/72339069014638637
    Start Timestamp: Wed Feb 24 09:06:47 GMT +0000 2021
    Idle Timestamp: Wed Feb 24 09:06:47 GMT +0000 2021
    Session-Timeout: 86400 seconds
    Idle-Timeout: 7200 seconds
    Session Statistics:
        Ingress: 0 packets 0 bytes
        Egress: 0 packets 0 bytes
    LIF Statistics:
        Ingress: 14 packets 896 bytes
        Egress: 0 packets 0 bytes
    Egress Class (Queue) Statistics:
        class-0: 0 packets 0 bytes
        class-1: 0 packets 0 bytes
        class-2: 0 packets 0 bytes
        class-3: 0 packets 0 bytes
        class-4: 0 packets 0 bytes
        class-5: 0 packets 0 bytes
        class-6: 0 packets 0 bytes
        class-7: 0 packets 0 bytes
    Ingress Policer Statistics:
        Level 1: 0 packets 0 bytes
        Level 2: 0 packets 0 bytes
        Level 3: 0 packets 0 bytes
        Level 4: 0 packets 0 bytes
```

#### Session Counters

Per default, the session counters are calculated using the LIF statistics which include all traffic received on the logical interface (InLIF) and all traffic sent on the logical interface (OutLIF) except control traffic which is directly sent to the IFP. Ingress traffic sent to CPU or transit traffic dropped by policer is still counted whereas egress traffic dropped by QoS is not counted.

Alternatively, it is possible to use the sum of all policer counters for ingress session counters which

can be changed in the AAA configuration profile by setting the accounting ingress source to POLICER (default is LIF) to include only accepted transit traffic.

Using LIF in egress and POLICER in ingress results in a symmetric behavior where only accepted transit traffic is counted.

Following the list of the RtBrick class counter attributes.

| Attribute | Name | Description |
|---|---|---|
| 42 | Acct-Input-Octets | Incoming session bytes (uint32) |
| 43 | Acct-Output-Octets | Outgoing session bytes (uint32) |
| 47 | Acct-Input-Packets | Incoming session packets (uint32) |
| 48 | Acct-Output-Packets | Outgoing session packets (uint32) |
| 52 | Acct-Input-Gigawords | Acct-Input-Octets overflow counter (uint32) |
| 53 | Acct-Output-Gigawords | Acct-Output-Octets overflow counter (uint32) |

The internal 64 bit counters are split over the standard 32 bit octet and gigaword counters by using the most significant 32 bit as gigawords and the least significant 32 bits as octets.

| Internal 64 Bit Counter | |
|---|---|
| 32 Bit Acct-Input/Output-Gigawords | 32 Bit Acct-Input/Output-Octets |

**Class Counters**

The outgoing class counters are filled by queue counters which ensures that only traffic leaving the device is counted here.

Following the list of the RtBrick class counter attributes.

| Attribute | Name | Description |
|---|---|---|
| 26-50058-81 | RtBrick-Class-0-Packets-Out | Outgoing Class-0 packets (uint64) |
| 26-50058-82 | RtBrick-Class-0-Bytes-Out | Outgoing Class-0 bytes (uint64) |
| 26-50058-83 | RtBrick-Class-1-Packets-Out | Outgoing Class-1 packets (uint64) |
| 26-50058-84 | RtBrick-Class-1-Bytes-Out | Outgoing Class-1 bytes (uint64) |
| 26-50058-85 | RtBrick-Class-2-Packets-Out | Outgoing Class-2 packets (uint64) |
| 26-50058-86 | RtBrick-Class-2-Bytes-Out | Outgoing Class-2 bytes (uint64) |
| 26-50058-87 | RtBrick-Class-3-Packets-Out | Outgoing Class-3 packets (uint64) |
| 26-50058-88 | RtBrick-Class-3-Bytes-Out | Outgoing Class-3 bytes (uint64) |
| 26-50058-98 | RtBrick-Class-4-Packets-Out | Outgoing Class-4 packets (uint64) |
| 26-50058-90 | RtBrick-Class-4-Bytes-Out | Outgoing Class-4 bytes (uint64) |
| 26-50058-91 | RtBrick-Class-5-Packets-Out | Outgoing Class-5 packets (uint64) |

| Attribute | Name | Description |
|---|---|---|
| 26-50058-92 | RtBrick-Class-5-Bytes-Out | Outgoing Class-5 bytes (uint64) |
| 26-50058-93 | RtBrick-Class-6-Packets-Out | Outgoing Class-6 packets (uint64) |
| 26-50058-94 | RtBrick-Class-6-Bytes-Out | Outgoing Class-6 bytes (uint64) |
| 26-50058-95 | RtBrick-Class-7-Packets-Out | Outgoing Class-7 packets (uint64) |
| 26-50058-96 | RtBrick-Class-7-Bytes-Out | Outgoing Class-7 bytes (uint64) |

Those attributes will be encoded as subattributes (RFC2865) similar to broadband forum (BBF) line attributes to reduce the size of the RADIUS message. Counters with a zero value will be also excluded from the packet.

```
▼ AVP: t=Vendor-Specific(26) l=246 vnd=RtBrick Inc.(50058)
    Type: 26
    Length: 246
    Vendor ID: RtBrick Inc. (50058)
  ▶ VSA: t=RtBrick-Class-0-Packets-Out(81) l=10 val=000000000000000a
  ▶ VSA: t=RtBrick-Class-0-Bytes-Out(82) l=10 val=00000000000003e8
  ▶ VSA: t=RtBrick-Class-1-Packets-Out(83) l=10 val=000000000000000b
  ▶ VSA: t=RtBrick-Class-1-Bytes-Out(84) l=10 val=00000000000003e9
  ▶ VSA: t=RtBrick-Class-2-Packets-Out(85) l=10 val=000000000000000c
  ▶ VSA: t=RtBrick-Class-2-Bytes-Out(86) l=10 val=00000000000003ea
  ▶ VSA: t=RtBrick-Class-3-Packets-Out(87) l=10 val=000000000000000d
  ▶ VSA: t=RtBrick-Class-3-Bytes-Out(88) l=10 val=00000000000003eb
  ▶ VSA: t=RtBrick-Class-4-Packets-Out(89) l=10 val=000000000000000e
  ▶ VSA: t=RtBrick-Class-4-Bytes-Out(90) l=10 val=00000000000003ec
  ▶ VSA: t=RtBrick-Class-5-Packets-Out(91) l=10 val=000000000000000f
  ▶ VSA: t=RtBrick-Class-5-Bytes-Out(92) l=10 val=00000000000003ed
  ▶ VSA: t=RtBrick-Class-6-Packets-Out(93) l=10 val=0000000000000010
  ▶ VSA: t=RtBrick-Class-6-Bytes-Out(94) l=10 val=00000000000003ee
  ▶ VSA: t=RtBrick-Class-7-Packets-Out(95) l=10 val=0000000000000011
  ▶ VSA: t=RtBrick-Class-7-Bytes-Out(96) l=10 val=00000000000003ef
  ▶ VSA: t=RtBrick-Policer-L1-Packets-In(97) l=10 val=0000000000000015
  ▶ VSA: t=RtBrick-Policer-L1-Bytes-In(98) l=10 val=00000000000007d1
  ▶ VSA: t=RtBrick-Policer-L2-Packets-In(99) l=10 val=0000000000000016
  ▶ VSA: t=RtBrick-Policer-L2-Bytes-In(100) l=10 val=00000000000007d2
  ▶ VSA: t=RtBrick-Policer-L3-Packets-In(101) l=10 val=0000000000000017
  ▶ VSA: t=RtBrick-Policer-L3-Bytes-In(102) l=10 val=00000000000007d3
  ▶ VSA: t=RtBrick-Policer-L4-Packets-In(103) l=10 val=0000000000000018
  ▶ VSA: t=RtBrick-Policer-L4-Bytes-In(104) l=10 val=00000000000007d4
```

*Figure 83. Class Counters*

#### Policer Counters

The incoming policer attributes count all traffic accepted (colored green) by ingress policers with dedicated packet and byte values per level.

Following the list of the RtBrick policer counter attributes.

| Attribute | Name | Description |
|---|---|---|
| 26-50058-97 | RtBrick-Policer-L1-Packets-In | Incoming Policer L1 accepted packets (uint64) |
| 26-50058-98 | RtBrick-Policer-L1-Bytes-In | Incoming Policer L1 accepted bytes (uint64) |
| 26-50058-99 | RtBrick-Policer-L2-Packets-In | Incoming Policer L2 accepted packets (uint64) |
| 26-50058-100 | RtBrick-Policer-L2-Bytes-In | Incoming Policer L2 accepted bytes (uint64) |
| 26-50058-101 | RtBrick-Policer-L3-Packets-In | Incoming Policer L3 accepted packets (uint64) |
| 26-50058-102 | RtBrick-Policer-L3-Bytes-In | Incoming Policer L3 accepted bytes (uint64) |
| 26-50058-103 | RtBrick-Policer-L4-Packets-In | Incoming Policer L4 accepted packets (uint64) |
| 26-50058-104 | RtBrick-Policer-L4-Bytes-In | Incoming Policer L4 accepted bytes (uint64) |

Those attributes will be encoded as subattributes (RFC2865) similar to broadband forum (BBF) line attributes to reduce the size of the RADIUS message. Counters with a zero value will be also excluded from the packet.

### Service Classification and Accounting

The underlying virtual output queueing (VOQ) is a common and efficient architecture for high performance switches but brings some major limitations. One obvious limitation is that queues must be allocated in ingress and cannot be changed in egress as shown in the picture below.

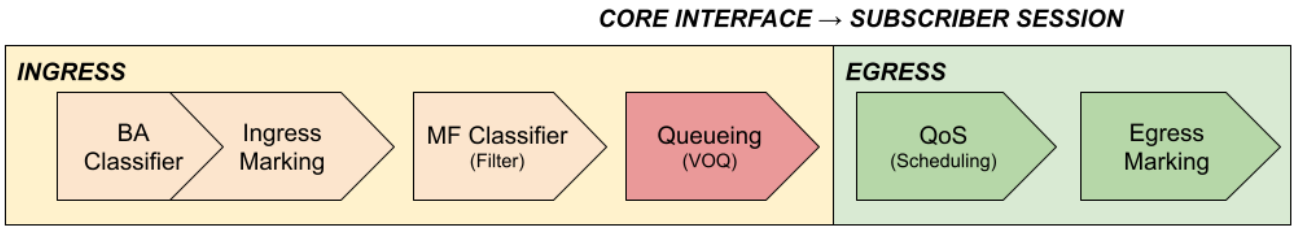



*Figure 84. Service Classification and Accounting*

Service classification and accounting is based on standard behavior aggregate (BA) and multifield classifier (MF) bound global or to ingress interfaces matching on some protocol fields assigning a traffic class. Downstream traffic (to subscriber) is counted per subscriber and class using queue counters to count only traffic not dropped by QoS. Upstream traffic (from subscribers) is countered per subscriber and policer level.

Some service providers would like to control if premium traffic is handled differently on a per subscriber basis which can be also achieved with VOQ as explained below.

Let us assume a product which handles video streaming services with higher priority or excludes those traffic from actual data volume. With VOQ the classification as Video is done on the ingress core interface equally for all subscribers regardless of the booked product. Now instead of changing the queue per subscriber in egress, the behavior of the queue can be changed to behave equally to

BestEffort which is supported in egress per subscriber.

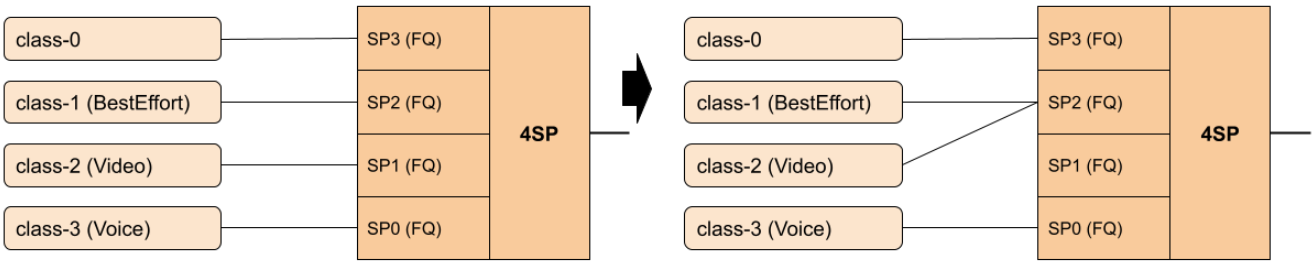



*Figure 85. Service Classification*

The corresponding Video traffic is now threaded equally to BestEffort but still counted separately. For accounting purposes the traffic of BestEffort and Video can be simply added in the billing infrastructure to get the actual amount of BestEffort traffic.

In upstream, the expected class and policer level can be assigned per subscriber because BA and MF classifiers are bound to the PPPoE session in ingress.

### 5.2.6. RADIUS Accounting Adjustment Configuration

The accounting adjustment allows to do some basic counter adjustment for RADIUS interims and stop accounting request messages using the following parameters from the AAA configuration profile (**global.access.aaa.profile.config**).

This counter adjustment allows normalizing counters with different encapsulations (double tagged, untagged, ...) to L3 counters for example.

| Parameter | Type | Description |
|---|---|---|
| accounting_ingress_source | uint8 | Source of session ingress counter (1: LIF or 3: POLICER)<br><br>*Default: LIF* |
| accounting_egress_source | uint8 | Source of session egress counter (1: LIF or 2: CLASS)<br><br>*Default: LIF* |
| ingress_byte_adjustment_value | float | Adjust ingress LIF counters by +/-N bytes per packet<br><br>*Default: 0.00* |
| ingress_byte_adjustment_factor | float | Adjust ingress LIF counters by factor<br><br>(executed after adjustment value)<br><br>*Default: 1.00* |

| Parameter | Type | Description |
|---|---|---|
| egress_byte_adjustment_value | float | Adjust egress LIF counters by +/-N bytes per packet<br><br>*Default: 0.00* |
| egress_byte_adjustment_factor | float | Adjust egress LIF counters by factor<br><br>(executed after adjustment value)<br><br>*Default: 1.00* |
| accounting_egress_class_byte_adjustment_value | float | Adjust egress CLASS counters by +/-N bytes per packet<br><br>*Default: 0.00* |
| accounting_egress_class_byte_adjustment_factor | float | Adjust egress POLICER counters by factor<br><br>(executed after adjustment value)<br><br>*Default: 1.00* |
| accounting_ingress_policer_byte_adjustment_value | float | Adjust ingress POLICER counters by +/-N bytes per packet<br><br>*Default: 0.00* |
| accounting_ingress_policer_byte_adjustment_factor | float | Adjust ingress LIF counters by factor<br><br>(executed after adjustment value)<br><br>*Default: 1.00* |

The byte adjustment value supports positive and negative values like -20.0 or 20.0. Provided decimal digits in the adjustment values are ignored.

The byte adjustment factors support positive values and only the first two decimal digits are used like 0.98 (-2%) or 1.02 (+2%).

### 5.2.7. RADIUS Redundancy

It is possible to configure multiple RADIUS authentication and accounting servers for redundancy and or load-balancing.

The following two algorithms are supported:

- **DIRECT (default):** The DIRECT algorithm is designed to primarily focus on **redundancy and fault tolerance**. In this mode, the RBFS subscriber daemon always attempts to send RADIUS requests to a single preferred server. If a response is received, the server continues to be used for subsequent requests. However, if the server becomes unresponsive or times out, the daemon fails over to the next server in the configured list. This process continues sequentially until a server responds or the list of servers is exhausted.

- **ROUND-ROBIN:** The ROUND-ROBIN algorithm implements **load-sharing** across all configured RADIUS servers. Each outgoing RADIUS request is sent to the next server in the list, circularly rotating through the servers. This allows the authentication and accounting load to be spread evenly across all available servers.

### 5.2.8. Test AAA

The test AAA subscriber feature allows operators to verify and test authentication and accounting (e.g. local or RADIUS) by emulating a subscriber without the need for external clients to be connected. This is a commonly used feature used during troubleshooting or to validate the RADIUS configuration.

Test subscribers will be created by adding a request object into the test aaa request table (**global.subscriber.1.test.aaa.request**) via API or CLI. This request object includes beside username and password also a spoofed interface, outer-vlan, inner-vlan and MAC address required to build corresponding attributes like NAS port identifiers or the vendor specific RtBrick-Access-Stack. It is also possible to add an Agent-Circuit-Id or Agent-Remote-Id to test line based authentication.

This new object will first trigger a validation plugin to reject invalid requests (e.g. subscriber-id out of range, missing mandatory attributes, ...). The add callback for this object will trigger the creation of the actual subscriber. Deleting the request object will trigger the termination of the subscriber. If this subscriber terminates for other reasons like CoA or CLI, the subscriber remains in a terminated state until the request object is deleted.

Compared to real subscribers, test subscribers will not trigger any forwarding state. Therefore no object will be created in the following tables for those subscribers:

- global.access.1.subscriber.ifl

But the following tables will be populated for operators to check dynamic QoS, filters or IGMP:

- global.access.1.subscriber.qos
- global.access.1.subscriber.qos.shaper
- global.access.1.subscriber.filter
- global.access.1.subscriber.igmp.service

Those entries are ignored internally as there is no matching subscriber interface (IFL) (global.access.1.subscriber.ifl). The only exception here is the filter table meaning that received filters (ADF) learned from RADIUS are installed in the ACL table and applied with the IP addresses of the test subscribers.

**Test AAA Subscriber-Id:**

In RBFS each subscriber is uniquely identified by subscriber-id which is a 64 bit number allocated in a distributed fashion by the application creating the subscriber (for example, pppoed for pppoe sessions). This uniqueness is achieved with the following format.

The first 8 bits are used to identify the application:

- 0x00 subscriberd
- 0x01 pppoed
- 0x02 l2tpd

The next 8 bits are used to identify the application sharding instance allocated (for example, pppoed.1, pppoed.2, and so on).

The remaining 48 bits are used to uniquely identify the subscriber, which theoretically allows up to 281474976710656 subscribers per application instance.

An emulated test subscriber needs also a subscriber-id.

```
-----
* 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

* +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

* | app-id (0) | app-instance | reserved |

* +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

* | subscriber

* +-+-+-+-+

-----
```

Assuming subscriber daemon instance 1 (subscriberd.1), the valid range for test subscriber identifiers is between 281474976710656 and 281479271677951.

```
Min Subscriber-Id: 281474976710656
0000 0000 0000 0001 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

Max Subscriber-Id: 281479271677951
0000 0000 0000 0001 0000 0000 0000 0000 1111 1111 1111 1111 1111 1111 1111 1111
```

The subscriber identifier will be allocated manually by operators from this range which has the advantage to easier identify the subscriber created by request.

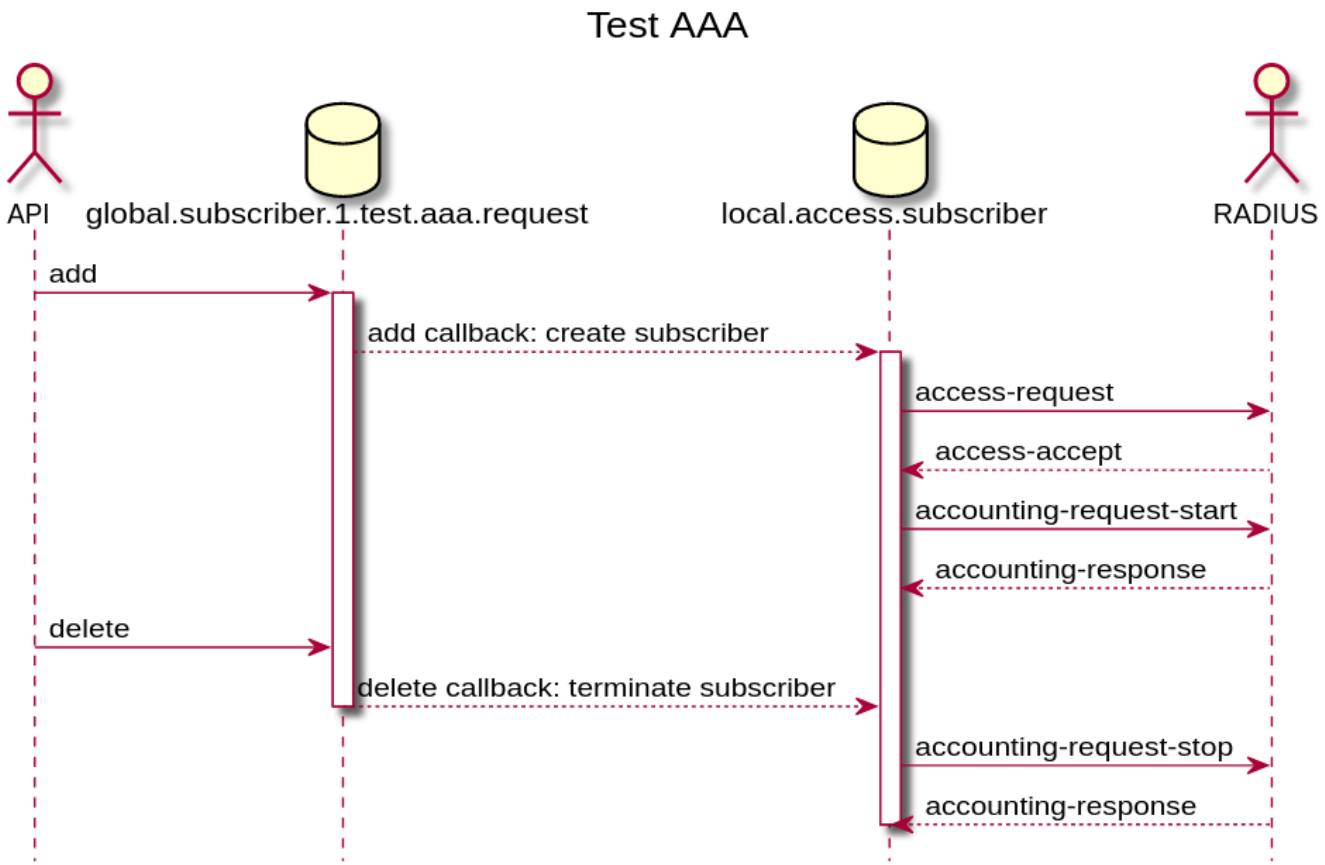



*Figure 86. Test AAA RADIUS Flow*

### Test AAA Attributes

| Attribute | Mandatory | API | CLI | Default |
|---|---|---|---|---|
| Subscriber-Id | Yes | subscriber_id | subscriber-id | |
| Access Profile | Yes | access_profile_name | access-profile | |
| AAA Profile | Yes | aaa_profile_name | aaa-profile | |
| Username | No | user_name | username | test |
| Password | No | password | password | test |
| Agent-Circuit-id | No | aci | aci | |
| Agent-Remote-id | No | ari | ari | |
| Interface (IFP) | * | interface_name | interface | ifp-0/0/0 |
| Outer VLAN | No | outer_vlan | outer-vlan | 0 |
| Inner VLAN | No | inner_vlan | inner-vlan | 0 |
| MAC Address | No | client_mac | ** | 00:00:00:00:00:00 |

** The interface is mandatory via CLI but optional via API.*

*** The client MAC address is currently supported via API only.*

### Test AAA via CLI

> In CLI, strings must not contain the “” character and therefore this is not permitted for username, aci or ari set via CLI. This limitation is valid for CLI only as subscribers created via API are allowed to use the “” character in the mentioned strings.

**Add Test Subscriber:**

```
supervisor@leaf1: op> test subscriber aaa request 281474976710656 pppoe-dual aaa-
default ifl-0/0/1 username user1@rtbrick.com aci "0.0.0.0/0.0.0.0 eth 0/0" ari
DEU.RTBRICK.01
```

**List All Test Subscribers:**

```
supervisor@leaf1: op> show subscriber filter type Test
Subscriber-Id          Interface        VLAN      State             Username
281474976710656        ifp-0/1/23       0:0       ESTABLISHED       user1@rtbrick.com
```

**Delete Test Subscriber:**

```
supervisor@leaf1: op> test subscriber aaa delete 281474976710656
```

### Test AAA via API

**Add Test Subscriber:**

```
{{rbfs_url}}/api/v1/rbfs/elements/{\{element}}/services/subscriberd.1/proxy/bds/object
/add
```

```
{
   "table": {
       "table_name": "global.access.1.test.aaa.request"
   },
   "objects": [
       {
           "attribute": {
               "subscriber_id": 281474976710658,
               "aaa_profile_name": "aaa-default",
               "access_profile_name": "pppoe-dual",
               "user_name": "user@rtbrick.com",
               "ari": "DEU.RTBRICK.01",
               "aci": "0.0.0.0/0.0.0.0 eth 0/0"
           }
       }
```

```
   ]
}
```

**Delete Test Subscriber:**

```
{{rbfs_url}}/api/v1/rbfs/elements/{\{element}}/services/subscriberd.1/proxy/bds/object
/delete
```

```
{
   "table": {
       "table_name": "global.access.1.test.aaa.request"
   },
   "objects": [
       {
           "attribute": {
               "subscriber_id": 281474976710658
           }
       }
   ]
}
```

**List All Test Subscribers:**

```
{{rbfs_url}}/api/v1/rbfs/elements/{\{element}}/services/subscriberd.1/proxy/bds/table/
walk
```

```
{
   "table": {
       "table_name": "global.access.1.test.aaa.request"
   }
}
```

### 5.2.9. Appendix A - RADIUS Dictionary

Following the RtBrick RADIUS dictionary in the well known FreeRADIUS format.

```
# This dictionary applies to RtBrick Full Stack (RBFS)
# https://www.rtbrick.com/
#
VENDOR          RtBrick 50058

BEGIN-VENDOR    RtBrick

ATTRIBUTE       RtBrick-Access-Hostname              21  string
ATTRIBUTE       RtBrick-Access-Port                  22  string
```

```
ATTRIBUTE       RtBrick-Access-Stack                23  string
ATTRIBUTE       RtBrick-Access-MAC-Address          24  string

ATTRIBUTE       RtBrick-Subscriber-Id               25  integer64
ATTRIBUTE       RtBrick-Subscriber-Ifl              26  string

ATTRIBUTE       RtBrick-Terminate-Code              27  integer

ATTRIBUTE       RtBrick-DHCPv4-Client-Options       30  octets
ATTRIBUTE       RtBrick-DHCPv4-Server-Options       31  octets
ATTRIBUTE       RtBrick-DHCPv6-Client-Options       32  octets
ATTRIBUTE       RtBrick-DHCPv6-Server-Options       33  octets

ATTRIBUTE       RtBrick-L2TP-Pool                   40  string
ATTRIBUTE       RtBrick-L2TP-Profile                41  string
ATTRIBUTE       RtBrick-L2TP-Tx-Connect-Speed       42  integer
ATTRIBUTE       RtBrick-L2TP-Rx-Connect-Speed       43  integer

ATTRIBUTE       RtBrick-QoS-Profile                 62  string
ATTRIBUTE       RtBrick-QoS-Shaper                  63  string
ATTRIBUTE       RtBrick-QoS-Parent-Scheduler        64  string
ATTRIBUTE       RtBrick-QoS-Policer                 65  string
ATTRIBUTE       RtBrick-QoS-MFC                     66  string
ATTRIBUTE       RtBrick-QoS-Queues                  67  string

ATTRIBUTE       RtBrick-AAA-Profile                 69  string
ATTRIBUTE       RtBrick-Service-Profile             70  string

ATTRIBUTE       RtBrick-IGMP-Status                 71  integer
ATTRIBUTE       RtBrick-IGMP-Profile                72  string
ATTRIBUTE       RtBrick-IGMP-Version                73  integer
ATTRIBUTE       RtBrick-IGMP-Max-Members            74  integer

ATTRIBUTE       RtBrick-HTTP-Redirect-URL           75  string
ATTRIBUTE       RtBrick-IPv4-ACL-In                 76  string
ATTRIBUTE       RtBrick-IPv4-ACL-Out                77  string
ATTRIBUTE       RtBrick-IPv6-ACL-In                 78  string
ATTRIBUTE       RtBrick-IPv6-ACL-Out                79  string

ATTRIBUTE       RtBrick-Class-0-Packets-Out         81  integer64
ATTRIBUTE       RtBrick-Class-0-Bytes-Out           82  integer64
ATTRIBUTE       RtBrick-Class-1-Packets-Out         83  integer64
ATTRIBUTE       RtBrick-Class-1-Bytes-Out           84  integer64
ATTRIBUTE       RtBrick-Class-2-Packets-Out         85  integer64
ATTRIBUTE       RtBrick-Class-2-Bytes-Out           86  integer64
ATTRIBUTE       RtBrick-Class-3-Packets-Out         87  integer64
ATTRIBUTE       RtBrick-Class-3-Bytes-Out           88  integer64
ATTRIBUTE       RtBrick-Class-4-Packets-Out         89  integer64
ATTRIBUTE       RtBrick-Class-4-Bytes-Out           90  integer64
ATTRIBUTE       RtBrick-Class-5-Packets-Out         91  integer64
ATTRIBUTE       RtBrick-Class-5-Bytes-Out           92  integer64
```

```
ATTRIBUTE       RtBrick-Class-6-Packets-Out          93  integer64
ATTRIBUTE       RtBrick-Class-6-Bytes-Out            94  integer64
ATTRIBUTE       RtBrick-Class-7-Packets-Out          95  integer64
ATTRIBUTE       RtBrick-Class-7-Bytes-Out            96  integer64

ATTRIBUTE       RtBrick-Policer-L1-Packets-In        97  integer64
ATTRIBUTE       RtBrick-Policer-L1-Bytes-In          98  integer64
ATTRIBUTE       RtBrick-Policer-L2-Packets-In        99  integer64
ATTRIBUTE       RtBrick-Policer-L2-Bytes-In         100  integer64
ATTRIBUTE       RtBrick-Policer-L3-Packets-In       101  integer64
ATTRIBUTE       RtBrick-Policer-L3-Bytes-In         102  integer64
ATTRIBUTE       RtBrick-Policer-L4-Packets-In       103  integer64
ATTRIBUTE       RtBrick-Policer-L4-Bytes-In         104  integer64

ATTRIBUTE       RtBrick-DNS-Primary                 131  ipaddr
ATTRIBUTE       RtBrick-DNS-Secondary               132  ipaddr
ATTRIBUTE       RtBrick-DNS-Primary-IPv6            134  ipv6addr
ATTRIBUTE       RtBrick-DNS-Secondary-IPv6          135  ipv6addr

ATTRIBUTE       RtBrick-Instance                    137  string
ATTRIBUTE       RtBrick-Gateway-Ifl                 138  string

ATTRIBUTE       RtBrick-Connection-Status-Message   139  string

ATTRIBUTE       RtBrick-LI-Action                   140  integer encrypt=2
ATTRIBUTE       RtBrick-LI-Identifier               141  integer encrypt=2
ATTRIBUTE       RtBrick-LI-Direction                142  integer encrypt=2
ATTRIBUTE       RtBrick-LI-MED-Instance             143  string  encrypt=2
ATTRIBUTE       RtBrick-LI-MED-IP                   144  ipaddr  encrypt=2
ATTRIBUTE       RtBrick-LI-MED-Port                 145  integer encrypt=2

#         VALUE MAPS
#
#       Attribute               Name        Number
VALUE   RtBrick-LI-Action       NOOP        0
VALUE   RtBrick-LI-Action       ON          1
VALUE   RtBrick-LI-Action       OFF         2

VALUE   RtBrick-LI-Direction    INGRESS     1
VALUE   RtBrick-LI-Direction    EGRESS      2
VALUE   RtBrick-LI-Direction    BOTH        3

VALUE   RtBrick-IGMP-Status     DISABLED    0
VALUE   RtBrick-IGMP-Status     ENABLED     1

VALUE   RtBrick-IGMP-Version    V1          1
VALUE   RtBrick-IGMP-Version    V2          2
VALUE   RtBrick-IGMP-Version    V3          3

END-VENDOR RtBrick
```

- The **RtBrick-DHCP-Client-Options** and **RtBrick-DHCPv6-Client-Options** attributes enable the transmission of DHCPv4/DHCPv6 options in the Access-Accept message as payload. These options are added to packets sent to the client, operating in relay/proxy mode or server mode.
- The **RtBrick-DHCP-Server-Options** and **RtBrick-DHCPv6-Server-Options** attributes are only supported in relay/proxy mode and are used to add options in packets sent toward the DHCP server.

### 5.2.10. Appendix B - RADIUS Standard Attributes Supported

**Disclaimer Note**: The following table presents the current development state of the RADIUS attributes supported in RBFS. Some of these may be incomplete and are subjected to change at any time without any notice.

| RFC | Attribute | | Access | | Accounting | | | | CoA | |
|---|---|---|---|---|---|---|---|---|---|---|
| | # | Name | Request | Response | Acct-Start | Acct-Interim | Acct-Stop | Acct-On | Request | Response |

2865

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Station-Id | | | | | | | | |
| | 32 | NAS-Identifier | X | | X | X | X | X | | |
| | 61 | NAS-Port-Type | X | | X | X | X | | | |
| 2866 | 40 | Acct-Status-Type | | | X | X | X | X | | |
| | 41 | Acct-Delay-Time | | | | | | | | |
| | 42 | Acct-Input-Octets | | | | X | X | | | |
| | 43 | Acct-Output-Octets | | | | X | X | | | |
| | 44 | Acct-Session-ID | X | | X | X | X | | | |
| | 45 | Acct-Authentic | | | | | | | | |
| | 46 | Acct-Session-Time | | | | | X | | | |
| | 47 | Acct-Input-Packets | | | | X | X | | | |
| | 48 | Acct-Output-Packets | | | | X | X | | | |
| | 49 | Acct-Terminate-Cause | | | | | X | | | |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 2869 | 52 | Acct-Input-Gigawords | | | | X | X | | | |
| | 53 | Acct-Output-Gigawords | | | | X | X | | | |
| | 55 | Event-Timestamp | | | X | X | X | | | |
| 2865 | 60 | CHAP-Challenge | X | | | | | | | |
| 2868 | 64 | Tunnel-Type | | X | X | X | X | | | |
| | 65 | Tunnel-Medium-Type | | X | | | | | | |
| | 66 | Tunnel-Client-Endpoint | | X | X | X | X | | | |
| | 67 | Tunnel-Server-Endpoint | | X | X | X | X | | | |
| | 69 | Tunnel-Password | | X | | | | | | |
| | 81 | Tunnel-Private-Group-ID | | X | X | X | X | | | |
| | 82 | Tunnel-Assignment-ID | | X | X | X | X | | | |
| | 83 | Tunnel-Preference | | X | | | | | X | |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 2869 | 85 | Acct-Interim-Interval | X | | | | | | X | |
| | 87 | NAS-Port-Id | X | | X | X | X | | | |
| | 88 | Framed-Pool | | X | | | | | | |
| 2868 | 90 | Tunnel-Client-Auth-ID | | X | X | X | X | | | |
| | 91 | Tunnel-Server-Auth-ID | | X | X | X | X | | | |
| 3162 | 97 | Framed-IPv6-Prefix | | X | X | X | X | | | |
| | 99 | Framed-v6-Route | | X | X | X | X | | | |
| | 100 | Framed-IPv6-Pool | | X | | | | | | |
| 3576 | 101 | Error-Cause | | | | | | | | X |
| 4818 | 123 | Delegated-IPv6-Prefix | | X | X | X | X | | | |
| 6911 | 171 | Delegated-IPv6-Prefix-Pool | | X | | | | | | |

4679

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Downstream | | | | | | | | |
| 26-3561-144 | Access-Loop-Encapsulation | X | X | X | X | X | | | |

| | |
|---|---|
| RtBrick | |
| 686 | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 50058-138 | Gateway-Ifl | | | | | | | | |
| 26-50058-139 | RtBrick-Connection-Status-Message | | X | | | | | X | |
| 26-50058-140 | RtBrick-LI-Action | | X | | | | | X | |
| 26-50058-141 | RtBrick-LI-Identifier | | X | | | | | X | |
| 26-50058-142 | RtBrick-LI-Direction | | X | | | | | X | |
| 26-50058-143 | RtBrick-LI-MED-Instance | | X | | | | | X | |
| 26-50058-144 | RtBrick-LI-MED-IP | | X | | | | | X | |
| 26-50058-145 | RtBrick-LI-MED-Port | | X | | | | | X | |

### 5.2.11. Appendix C - Access Deployments

The following picture shows different deployments typically used in access networks.

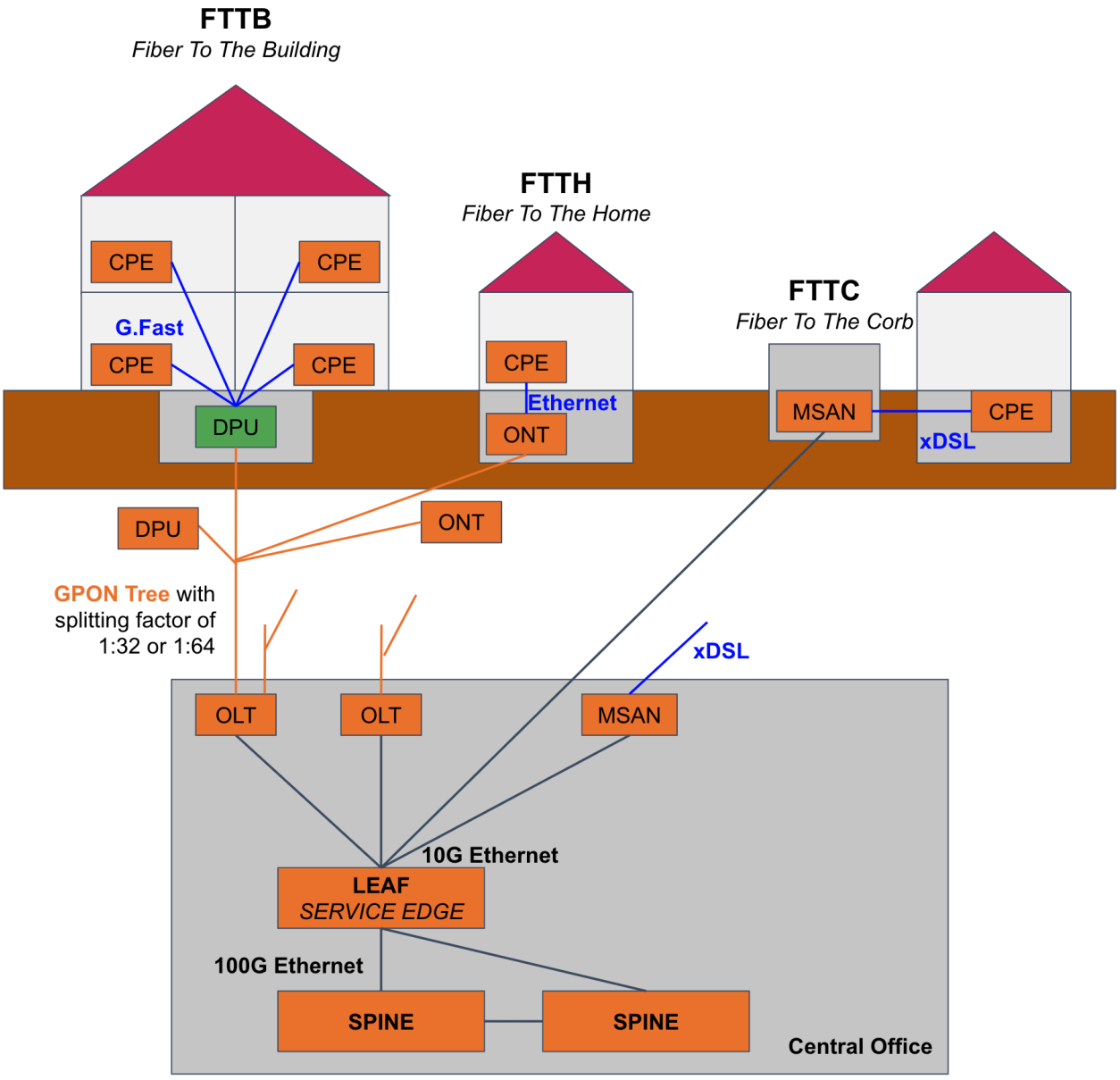



*Figure 87. Access Deployments*

The LEAF and SPINE are bare metal switches running RBFS where the LEAF is responsible for service edge functionalities like PPPoE subscriber termination and the SPINE builds the actual fabric core. Multiple LEAF switches will be connected to a pair of spine switches.

# 5.3. RBFS Carrier-Grade Network Address Translation

## 5.3.1. Carrier-Grade Network Address Translation Overview

### Carrier-Grade NAT

The rise of technologies and the rapid growth of mobile devices and cloud services worldwide resulted in the exhaustion of IPv4 addresses. Broadband service providers face a growing challenge as this 32-bit address is insufficient to meet the ever-growing demand and the cost of obtaining public IP addresses for every individual subscriber. Though the launch of IPv6 can address the IPv4 depletion problem, migrating to an IPv6 network becomes vastly complex and costly. IPv6 also does not provide backward compatibility with IPv4. Consequently, migrating to IPv6 is not an ideal

solution for many broadband service providers.

Carrier-Grade Network Address Translation (CGNAT) is a prominent technology solution that addresses the IPv4 exhaustion challenge of service providers. The solution helps to use private network addresses and to limit the use of publicly routable IPv4 addresses significantly.

CGNAT allows service providers to serve a large number of their subscribers using a limited number of public IPv4 addresses. It saves costs on public IPv4 addresses and conserves their IPv4 address pools.

**Static NAT**

Static NAT associates one private IP address with one public IP address. This is a one-to-one mapping and you must manually define the mapping between a private and public address.

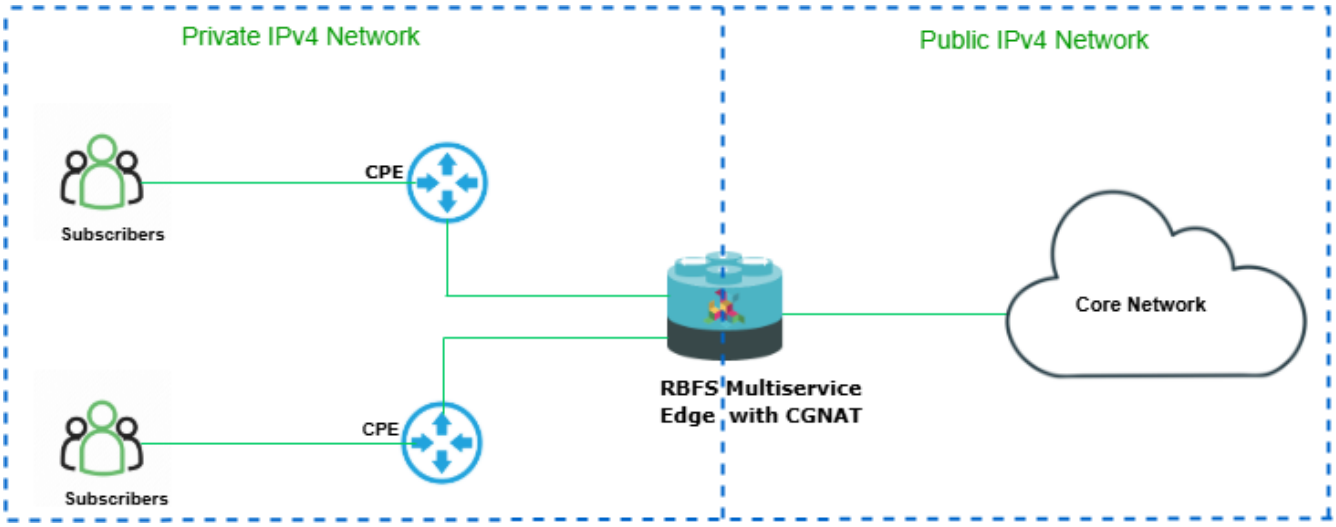
**Dynamic NAT**

In Dynamic NAT, internal private IPv4 addresses are mapped with a public IPv4 address and this mapping of a private IPv4 address to a public IPv4 address happens dynamically. The router dynamically picks an address from the address pool that is not currently assigned.

**NAT444**

CGNAT supports NAT444. NAT 444 refers to three sets of IPv4 addresses: Customer private, ISP private, and public Internet. Network address translation occurs from customer private to ISP private, and then ISP private to public. The NAT444, also known as CGNAT, functionality maps IPv4 subscriber private addresses to IPv4 public addresses.

The following diagram illustrates a high-level view of CGNAT.




**CGNAT Benefits**

CGNAT allows service providers to provide services to a large number of their subscribers using a limited number of public IPv4 addresses. The benefits include:

#### Conserve Public IPv4 Addresses

The solution helps service providers conserve expensive public IPv4 addresses by enabling multiple subscribers to share a single public IPv4 address. With CGNAT, one public IPv4 address can manage hundreds of devices within the private network. It reduces the requirement of continually buying additional public IP addresses.

#### Enhance Security

In addition to stretching the limited pool of public IPv4 addresses even further, CGNAT also provides significant security benefits. It helps to enhance network security by keeping the internal addressing private from the external network. It makes it difficult for attackers to target specific devices on the network by preventing attacks that target specific IP addresses.

#### Performance at Carrier-Grade

The CGNAT solution implements the functions of IPv4 address translation in the network of service providers. Service providers can deploy NAT in a way that allows multiple subscribers to share a single global IPv4 address and scale to several thousands of address translations. CGNAT operations are implemented at the forwarding plane (hardware level) to ensure optimal performance. This hardware-level implementation achieves high throughput with no performance impact and packet processing occurs without overburdening the CPU. The solution offers carrier-grade scalability with fast translation rates and a large number of IPv4 address and port number translations.

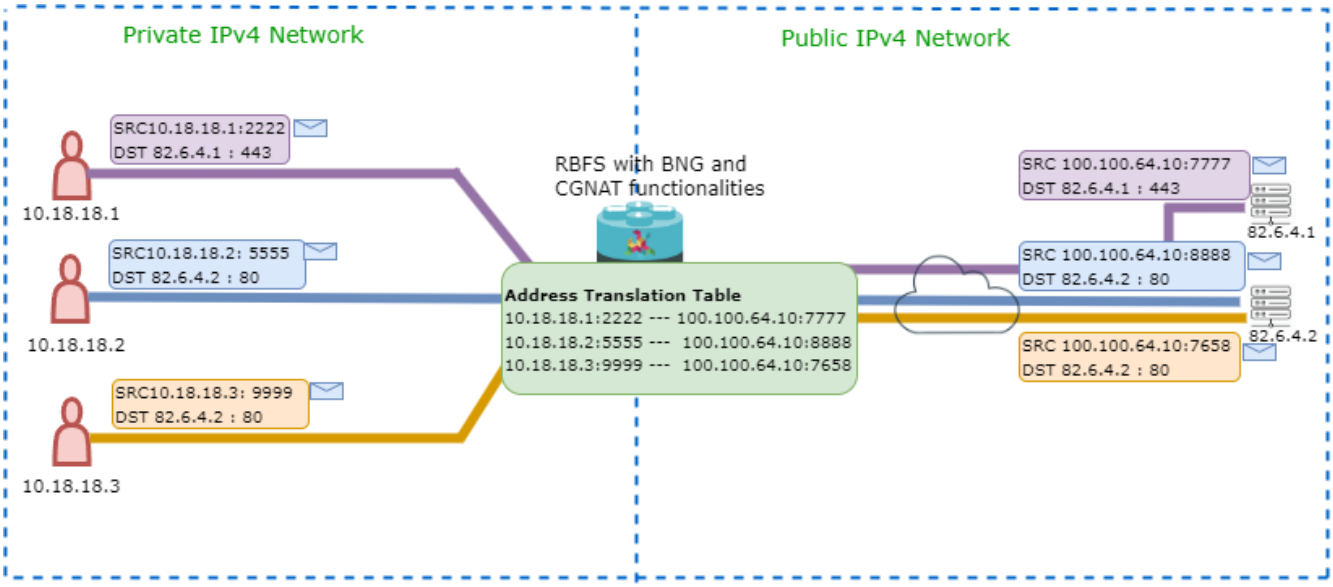
### Understanding CGNAT Implementation

CGNAT solution has been designed to support Port Address Translation, also known as Network Address Port Translation (NAPT), which has the greatest potential to conserve IPv4 addresses for service providers. NAPT is known as an effective method to allow multiple devices to connect to the Internet using a single public IPv4 address.

#### Network Address Port Translation

Network Address Port Translation is a dynamic NAT in which port numbers along with IP address are used to identify which traffic belongs to which private IP address. CGNAT translates the source private IPv4 address and port number to a public source IPv4 address and unique port number. This allows multiple devices with different private IP addresses to use a single public IPv4 address. The unique port number ensures that the traffic is delivered to the correct device.

The following diagrams illustrate how CGNAT works at a high level.




The diagram shows three subscribers using private IPv4 addresses (`10.18.18.1`, `10.18.18.2`, and `10.18.18.3` send traffic to two different servers (`82.6.4.1` and `82.6.4.2`) on the public network.

BNG with CGNAT performs the address translation by replacing the source private IPv4 address with the public IPv4 address from the address pool and the source port number with a unique port number. After the address translation, the device forwards the traffic to the destination servers. After the translation, packets have a new source IPv4 address which is the public IPv4 and a unique port number as per the mapping in the address translation table.

CGNAT maintains a translation table with the entries. Entries are records of mapped private IPv4 addresses and port numbers with the public IPv4 addresses and port numbers.

The downstream traffic from the servers traverses to the CGNAT device. The packets coming from external hosts include the destination address as the (translated) public IPv4 address. CGNAT device performs reversal address translation for these packets as per the translation table mapping for the downstream traffic.

NAPT helps to significantly reduce the number of logs generated as it generates logs only during the allocation and release of each block of ports.

**Deterministic NAT**

CGNAT offers support for deterministic NAT mode, which provides a consistent mapping of private IPv4 addresses with public IPv4 addresses and port ranges. This mode ensures a one-to-one mapping of private IPv4 addresses with public IPv4 addresses, allowing you to specify the private address and its matching public address and port range. The given private IPv4 address is always translated to the same public address.

In addition, deterministic NAT guarantees a predetermined and fixed translation for a given internal address with a public IPv4 and port combination, ensuring consistency and stability in the mapping. This feature is particularly useful for applications that involve security protocols, as well as for service providers who need to track subscriber sessions.

Furthermore, deterministic NAT significantly reduces address translation logs, as private IPv4 addresses are always mapped to public IPv4 addresses and port ranges.

#### Address Translation Table

Port mapping is a feature that allows multiple devices to share the same public IPv4 address with different port numbers. CGNAT generates a unique port number for each subscriber session. Port mapping helps to determine the correct host among the many devices in the private network that use the same public IPv4 address.

CGNAT maintains an address translation table that maps private IPv4 addresses and port numbers to their corresponding public IPv4 addresses and ports. Whenever an incoming packet arrives, CGNAT checks the translation table to see if a translation entry exists for that packet. If there's an entry, the CGNAT replaces the source IPv4 address and port number in the packet header with the mapped public IPv4 address and port number.

When a device on the Internet sends packets downstream, the CGNAT software uses the translation table for address translation reversal. It replaces the destination public IPv4 address and port number in the packet header with the corresponding private IPv4 address and port number.

#### NAT IP Pools and Chaining

A NAT pool contains a range of multiple public IP addresses. You can create multiple pools and associate them with a NAT service. Pool chaining is a method in which you can associate one pool with another. For a pool, you can define 'next pool name'; so that when the pool gets exhausted with the IPv4 addresses, the next pool that is defined will take over.

#### Ageing

Ageing refers to the time that a translation entry exists or remains in the address translation table after it was last used.

The entries in the translation table have a finite lifespan as the software implements mechanisms to handle session timeouts.

When a host sends a packet to a destination, CGNAT translates the private IPv4 address and port to a public IPv4 address and port. This mapping is recorded as an entry in the address translation table. The software always looks up the translation table whenever it receives a packet to verify that any entry exists for the packet in the translation table. The software performs the address translation based on the recorded entries and its associated mappings for both inbound and outbound packets.

If there is no activity related to a specific mapping (for example, when there are no incoming or outgoing packets), the mapping will eventually be removed from the address translation table, once idle or unused entries are detected, entries removed from the address translation table to free up resources for the new upcoming traffic flows.

For TCP, idle or unused flows are typically detected by the receipt of TCP FIN-ACK packets, which indicate that both sides of the connection have finished sending data. Once detected, these flows are removed after a configurable aging timer expires.

For UDP, as it is a connectionless protocol, idle or unused flows are detected by periodically polling the flow traffic. As the number of UDP traffic flows increases, the idle flow detection mechanism also increases, ensuring that idle flows are removed in a timely manner. The aging of UDP flows is

proportional to the number of traffic flows.

#### Ageing Timeout for UDP-based DNS Traffic

You can define an ageing timeout that controls how long the system maintains state information for UDP-based DNS traffic before considering the flow inactive and clearing it from the NAT table.

UDP traffic has a default ageing timeout of 30 seconds in NAT implementation.

> The NAT ageing timeout is 300 seconds for TCP and UDP traffic. The actual time taken to flush aged-out NAT rules depends on the number of NAT rules and the number of counters.

#### Support for IPoE N:1 Shared VLAN in Single VRF

CGNAT is supported for the IPoE N:1 shared VLAN model within the same VRF instance.

#### Logging

RBFS provides traffic monitoring capabilities to track and log network activities. You can enable logging for CGNAT operations.

> Logging is not completely supported on CGNAT.

#### NAT for ICMP Messages

The NAT-enabled RBFS devices can perform network address translation for ICMP echo request and reply messages. Subscribers can use ping command (ICMP Echo Request, Echo Reply) to check internet connectivity to external internet addresses when NAT is enabled. This feature is enabled by default.

#### NAT for TCP Control Messages

RBFS with NAT supports address translation for TCP control messages using ACLs. It includes the TCP control messages such as FIN, FIN-ACK, RST, RST-ACK, SYN, and SYN-ACK. This feature is enabled by default.

#### CGNAT Support for Traceroute

The RBFS CGNAT provides support for Traceroute and critical ICMP error handling on the Edgecore AGR420, UfiSpace S9600-72XC, and UfiSpace S9600-102XC platforms. This feature ensures that diagnostic tools can function correctly across NAT boundaries by accurately translating and forwarding "TTL Exceeded" and "Fragmentation Needed" messages.

#### NAT-miss Traffic Handling in Control Plane on Q2C Platforms

RBFS supports a mechanism to handle NAT-miss packets at control plane. A NAT-miss occurs when a downstream packet arrives at a device configured for address translation, but no matching NAT rule exists for that flow. A new ACL rule issued to detect such packets based on NAT-miss status and destination prefix match. These packets are redirected to the control plane, which performs network address translation in software and forwards them.

#### Support for Path MTU Discovery on CGNAT Deployments

Path MTU Discovery works in NAT environments for PPPoE and IPoE subscribers. RBFS with CGNAT-enabled can send ICMP "Fragmentation Needed" messages to the source when packet fragmentation is required, allowing the source to adjust the packet size accordingly. It allows for dynamic MTU adjustment and prevent packet drops and fragmentation related issues.

Limitations:

- NAT for fragmented IPv4 packets that lack TCP or UDP headers is not supported.
- Fragmentation for NAT subscriber traffic is not supported.

#### CGNAT N:1 with ACL-Based NAT Control

In a BNG N:1 deployment, CGNAT by default is enabled or disabled for all subscribers in the same VLAN as configured in the NAT profile. If, however, it is required to have a mix of NAT and non-NAT subscriber in a VLAN, then NAT needs to be enabled for all subscribers in the VLAN and an ACL needs to be configured using the IPv4 prefix of each subscriber or set of subscribers for whom the NAT needs to be disabled, to override the NAT profile configuration from getting applied.

**NAT Enabled Subscribers**:

By default, all subscribers undergo address translation via CGNAT. However, specific subscribers can be configured to bypass NAT using ACL rules. This allows selected subscribers to be forwarded without translation.

**Non-NATed Subscribers**:

All subscribers are, by default, forwarded without NAT when the service profile is configured as non-NATed. In this mode, selective NAT enablement for some specific subscribers through ACLs is not supported. Subscribers designated as non-NATed remain excluded from NAT processing.

> You can configure ACLs to exclude specific subscribers from NAT in a NAT-enabled environment. You cannot include specific subscribers for NAT in a non-NATed environment.

**Example: NAT Bypass for Specific Subscribers**

In a CGNAT N:1 deployment, subscribers within the same VLAN (for example, VLAN 100) and in the 198.51.100.0/24 range are configured with NAT enabled by default. Under normal operation, all subscriber traffic in this VLAN is translated using CGNAT.

In this scenario, specific subscribers (198.51.100.10 and 198.51.100.20) in the same VLAN require direct internet access without NAT. To meet this requirement, an ACL is configured to match these subscriber IP addresses. The ACL overrides the default NAT behavior for the matched subscribers within the VLAN and bypasses translation. As a result, traffic from 198.51.100.10 and 198.51.100.20 is forwarded without NAT, while all other subscribers in the VLAN continue to be NATed. This approach enables selective NAT bypass within a shared VLAN without modifying the overall service profile.

For configuration information, see the Selective NAT Bypass for Subscribers Using ACLs section.

#### Guidelines and Limitations

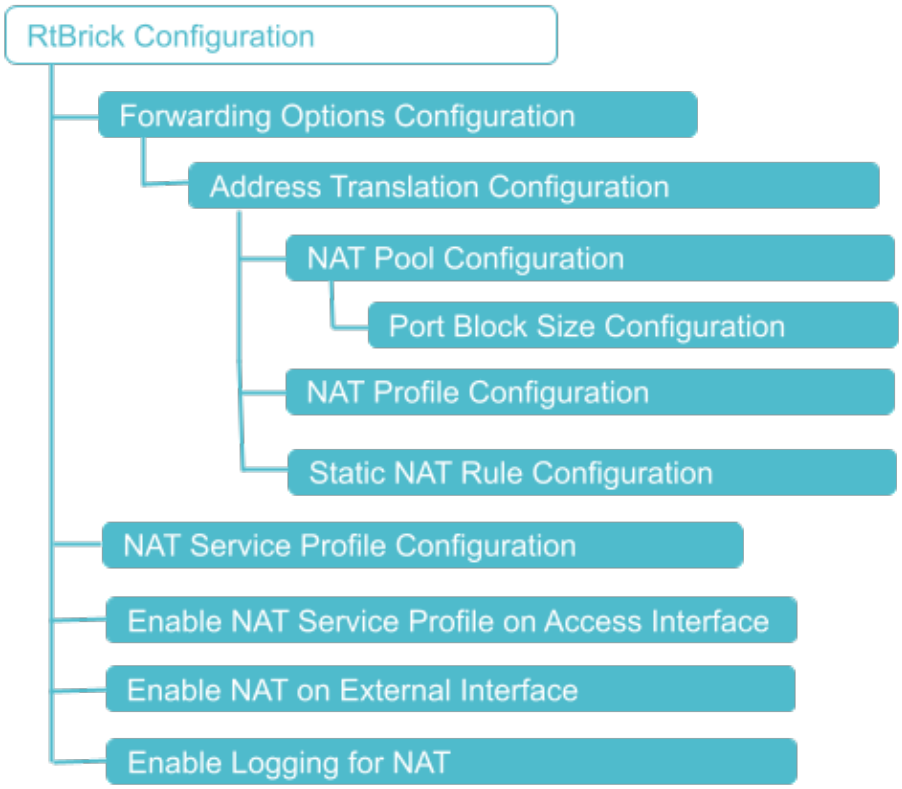
- On Q2A platforms, avoid installing IP forwarding routes that overlap with the public IP address range used for CGNAT.

#### Supported Platforms

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

### 5.3.2. Carrier-Grade NAT Configuration

#### Configuration Hierarchy




#### Prerequisites

Before configuring CGNAT, ensure that the appropriate platform profile is configured on the device. The platform profile determines the operating mode of the system and enables the required hardware resources for NAT functionality.

Configure one of the following platform profiles based on the deployment type:

- Use `nat_1q` or `nat_4q` profile when the system operates as a RBFS CGNAT, where NAT is integrated with subscriber management.
- Use this `nat_standalone` profile when the system operates as a dedicated CGNAT appliance without BNG subscriber functionality.

The platform profile must be configured and the system must be rebooted before proceeding with the CGNAT configuration.

Reboot the system for the changes to take effect, after setting the platform profile. For information about Platform Profile configuration, see Platform Profile.

### CGNAT Configuration

You must perform the following tasks to configure CGNAT.

1. Configure NAT Pool
2. Configure NAT Profile
3. Configure NAT Rule
4. NAT Service Profile Configuration
5. Enable NAT on an Instance
6. Enable NAT Service Profile on Access Interface
7. Enable NAT on External Interface
8. Enable Logging for NAT

### Configuration Syntax and Commands

The following sections describe the CGNAT configuration syntax and commands.

#### Configuring NAT Profile

A NAT profile defines how the NAT device has to perform the IPv4 address translation. NAT profile allows you to define an instance, IPv4 address pools, maximum number of translations, and mapping a particular internal IPv4 address with a particular external IPv4 address for a deterministic address translation.

A single NAT profile can be attached to subscribers across different instances. This means that regardless of which instance a subscriber belongs to, they can share the same NAT profile. For example, subscribers in different routing instances can be managed using the same NAT settings.

Subscribers within the same instance can be assigned different NAT profiles. Different groups of subscribers or services within the same instance can have tailored NAT configurations.

The configuration limit of NAT rules per subscriber in NAT profile is 16,384.

**Syntax:**

**set forwarding-options address-translation profile** <profile-name> <attribute> <value>

| Attribute | Description |
|---|---|
| `<profile-name>` | Specify the NAT profile name. |

| Attribute | Description |
| --- | --- |
| `deterministic [true]` | Specify deterministic as true to enable deterministic NAT for the profile. Deterministic NAT allows subscribers always to connect with a single public IP. |
| `ip-protocol` | Specify the protocol: tcp, udp, or other. |
| `ip-protocol ageing-timeout <ageing-timeout>` | Specify the ageing time value for the protocol. Default, 300 seconds for TCP and UDP. The allowed Ageing Time Out range values are as follows:<br>TCP: 60 - 86400 seconds<br>UDP: 120 - 86400 seconds |
| `ip-protocol control-plane-ageing-timeout <ageing-timeout>` | Specify the TCP/UDP control plane ageing timeout value. The default values are as follows:<br><br>TCP-control-plane: 10 seconds<br>UDP-control-plane: 30 seconds<br><br>The allowed Ageing Time Out range values are as follows:<br><br>TCP Control-plane: 5 - 120<br>UDP Control-plane: 5 - 120<br>Other: 120 - 86400 |
| `max-rules` | Specify the maximum number of rules per subscriber or an interface. The supported values for maximum address translation per subscriber or an interface are 64, 128, 256, 512, 1024, 2048, 4096, 8192, and 16384. |
| `pool` | Specify the name of the public IP address pool. |

The following commands configure the NAT profile named `nat_profile1`. The nat profile `nat_profile1` is configured on the instanced `vrf1` with a pool attached `nataddr_pool1`. Maximum rules are configured as 128 rules and the aging period is configured as 600 seconds for TCP traffic, 300 seconds for UDP traffic, and 300 for 'other' traffic.

```
set forwarding-options address-translation profile nat_profile1
set forwarding-options address-translation profile nat_profile1 pool nataddr_pool1
set forwarding-options address-translation profile nat_profile1 max-rules 128
set forwarding-options address-translation profile nat_profile1 ip-protocol tcp
ageing-timeout 600
set forwarding-options address-translation profile nat_profile1 ip-protocol udp
ageing-timeout 300
set forwarding-options address-translation profile nat_profile1 ip-protocol other
ageing-timeout 300
```

Example Configuration:

```
supervisor@rtbrick>cbng1.rtbrick.net: cfg> show config forwarding-options address-
```

```
translation profile
{
  "rtbrick-config:profile": [
    {
      "profile": "nat_profile1",
      "pool": "nataddr_pool1",
      "max-rules": "128",
      "ip-protocol": {
        "tcp": {
          "ageing-timeout": 600
        },
        "udp": {
          "ageing-timeout": 300
        },
        "other": {
          "ageing-timeout": 300
        }
      }
    }
  ]
}
```

#### NAT Pool Configuration

A NAT IP address pool includes a set of public IPv4 addresses that are used for network address translation. You can create multiple public IPv4 address pools and one pool includes a range of public IPv4 addresses. These pools allocate public IPv4 addresses to subscribers during address translation. While configuring a pool, you can define the group of public IPv4 addresses belonging to that pool by specifying the lowest and highest IP addresses.

The system allows you to create multiple pools and define the association among them. You can define the 'next-pool-name' that takes over when the current pool gets exhausted with the IPv4 addresses. When one pool gets exhausted, the next pool takes over and starts serving the IP addresses to subscribers when the address translation occurs.

> Static and dynamic IP pools cannot be configured simultaneously; they are mutually exclusive.

**Syntax:**

**set forwarding-options address-translation pool** <pool-name> <attribute> <value>

| Attribute | Description |
|---|---|
| `<pool-name>` | Specify the name of the address pool. |
| `ipv4-address` | Specify both the highest and lowest IPv4 addresses in the range of IPv4 addresses for the pool. |

| Attribute | Description |
| --- | --- |
| `ipv4-address high` | Specify the highest IPv4 address in the address pool. You must specify the highest IP address in the range of IP addresses. |
| `ipv4-address low` | Specify the lowest IPv4 address in the address pool. You must specify the lowest IP address in the range of IP addresses. |
| `next-pool-name` | Specify the name of the next address pool that is to be used when the current address pool is allocated completely. |

Example Configuration:

The following commands configure `nataddr_pool1` as the NAT pool and `nataddr_pool2` as the next pool.

It indicates NAT pool `nataddr_pool1` contains a range of public IPv4 addresses from `100.100.100.1` to `100.100.100.5`.

When the pool `nataddr_pool1` has fully allocated its IPv4 addresses, the next pool named `nataddr_pool2` will start allocating IPv4 addresses from its pool. The pool `nataddr_pool2` includes a rage of IPv4 address from `100.100.101.1` to `100.100.101.150`.

```
set forwarding-options address-translation pool nataddr_pool1
set forwarding-options address-translation pool nataddr_pool1 next-pool-name
nataddr_pool2
set forwarding-options address-translation pool nataddr_pool1 ipv4-address low
100.100.100.1
set forwarding-options address-translation pool nataddr_pool1 ipv4-address high
100.100.100.5
set forwarding-options address-translation pool nataddr_pool2 ipv4-address low
100.100.101.1
set forwarding-options address-translation pool nataddr_pool2 ipv4-address high
100.100.101.150
```

Example Configuration:

```
supervisor@rtbrick: cfg> show config forwarding-options address-translation pool
{
  "rtbrick-config:pool": [
    {
      "pool-name": "nataddr_pool1",
      "next-pool-name": "nataddr_pool2",
      "ipv4-address": {
        "low": "100.100.100.1",
        "high": "100.100.100.5"
      }
    },
    {
      "pool-name": "nataddr_pool2",
```

```
      "ipv4-address": {
        "low": "100.100.101.1",
        "high": "100.100.101.150"
      }
    }
  ]
}
```

#### NAT Rule Configuration

You can define NAT rules only for static NAT. A NAT rule defines a match condition and a corresponding action. After you specify NAT rules, each packet is matched with each NAT rule. If a packet matches the condition specified in a rule, then the action corresponding to that match occurs. Match rules govern how the translation of private IPv4 addresses to public IPv4 addresses is performed.

With NAT rules, you can define how address translation is applied to traffic, and how to handle various protocols and data traffic, such as TCP and UDP, to ensure proper address translation and the mappings of private addresses to public addresses.

Rules also define how to handle inbound and outbound traffic, different protocols, and data traffic such as TCP and UDP for ensuring the proper address translation of traffic.

Configuration limit for NAT maximum rules per subscriber is 16,384.

**Syntax:**

**set forwarding-options address-translation rule** <rule-name> <attribute> <value>

| Attribute | Description |
|---|---|
| `<rule-name>` | Specify the name of the rule. |
| `ordinal <ordinal-value>` | Specify the ordinal value. An ordinal value is a numerical representation that indicates its relative position or order. |
| `ordinal <ordinal-value> instance` | Specify the RBFS instance name. |
| `ordinal <ordinal-value> ip-protocol [tcp/udp]` | Specify the IP protocol, TCP or UDP. |
| `ordinal <ordinal-value> local [ipv4-address/port]` | Specify the private IPv4 address or port number that needs to be translated. |
| `ordinal <ordinal-value> public [ipv4-address/port]` | Specify the public IPv4 address. This public IP will be mapped with the private IP in the translation table. |

#### Configuring NAT Service Profile

You must create a NAT service profile and attach the service profile with the access interface for enabling CGNAT on the interface.

**Syntax:**

**set access service-profile** <profile-name> <attribute> <value>

| Attribute | Description |
|---|---|
| `<profile-name>` | Name of the service profile. |
| `address-translation profile <profile>` | Specify the profile name for the address translation. |

The following configuration defines a service profile named nat_service. It associates the NAT profile named 'nat_profile1' to the 'nat_service' profile. It indicates any subscriber using this service profile will use the configuration defined in nat_profile1.

```
set access service-profile nat_service
set access service-profile nat_service address-translation profile nat_profile1
```

Example:

```
{
  "rtbrick-config:service-profile": [
    {
      "profile-name": "nat_service",
      "address-translation": {
        "profile": "nat_profile1"
      },
      "qos": {
        "profile": "subs-triple-play"
      }
    }
  ]
}
```

#### Enable NAT Service Profile on the Access Interface

It is required to attach the NAT service profile to the access interface for enabling address translation on the interface.

**Syntax:**

**set access interface [double-tagged** | **single-tagged** | **untagged]** <interface-name> <outer-vlan-min> <outer-vlan-max> <inner-vlan-min> <inner-vlan-max> **service-profile-name** <service-profile-name>

| Attribute | Description |
|---|---|
| service-profile | Configure global service profile. |
| <outer-vlan-min> | Specify the minimum number of outer VLANs. Allowed range: 1 - 4094. |
| <outer-vlan-max> | Specify the maximum number of outer VLANs. Allowed range 1 - 4094. |
| <inner-vlan-min> | Specify the minimum number of inner VLANs. Allowed range: 1 - 4094. |
| <inner-vlan-max> | Specify the maximum number of inner VLANs. Allowed range 1 - 4094. |
| service-profile-name | Specify the name of the service profile. |

The following commands attach service profile `nat_service` to the interface (double-tagged) `ifp-0/0/17`. The configuration shows the range of outer VLANs as 1000 - 1007, and the range inner VLANs as 84 - 4084. The access type configured is IPoE and the service profile is NAT service. AAA profile name is `ipoe-aaa` and the gateway IFL is `lo-0/0/0/100`.

IPoE subscriber with outer VLAN between 1000 and 1007, and inner VLAN between 84 and 4084 will be matched with this NAT service profile and a corresponding action will be taken. Anything outside these vlans will not have any action from this NAT service profile.

```
set access interface double-tagged ifp-0/0/17 1000 1007 84 4084
set access interface double-tagged ifp-0/0/17 1000 1007 84 4084 access-type IPoE
set access interface double-tagged ifp-0/0/17 1000 1007 84 4084 access-profile-name
ipoe
set access interface double-tagged ifp-0/0/17 1000 1007 84 4084 service-profile-name
nat_service
set access interface double-tagged ifp-0/0/17 1000 1007 84 4084 aaa-profile-name ipoe-
aaa
set access interface double-tagged ifp-0/0/17 1000 1007 84 4084 gateway-ifl lo-
0/0/0/100
```

```
{
  "rtbrick-config:interface": {
    "double-tagged": [
      {
        "interface-name": "ifp-0/0/17",
        "outer-vlan-min": 1000,
        "outer-vlan-max": 1007,
        "inner-vlan-min": 84,
        "inner-vlan-max": 4084,
        "access-type": "IPoE",
        "access-profile-name": "ipoe",
        "service-profile-name": "nat_service",
        "aaa-profile-name": "ipoe-aaa",
```

```
        "gateway-ifl": "lo-0/0/0/100"
      }
    ]
  }
}
```

The following commands configure a double-tagged interface `ifp-0/0/6` and outer VLAN minimum value is 1000, maximum value as 1007, inner VLAN minimum value as 84 and maximum value as 4084. The access type configured is PPPoE, service profile is NAT service. AAA profile name is configured as `ipoe-aaa`.

It indicates PPPoE subscriber with outer VLAN between 1000 and 1007, and inner VLAN between 84 and 4084 will be matched with this NAT service profile and a corresponding action will be taken. Anything outside these vlans will not have any action from this NAT service profile.

```
set access interface double-tagged ifp-0/0/16 1000 1007 84 4084
set access interface double-tagged ifp-0/0/16 1000 1007 84 4084 access-type PPPoE
set access interface double-tagged ifp-0/0/16 1000 1007 84 4084 access-profile-name
pppoe
set access interface double-tagged ifp-0/0/16 1000 1007 84 4084 service-profile-name
nat_service
set access interface double-tagged ifp-0/0/16 1000 1007 84 4084 aaa-profile-name ipoe-
aaa
```

Example:

```
{
  "rtbrick-config:interface": {
    "double-tagged": [
      {
        "interface-name": "ifp-0/0/16",
        "outer-vlan-min": 1000,
        "outer-vlan-max": 1007,
        "inner-vlan-min": 84,
        "inner-vlan-max": 4084,
        "access-type": "PPPoE",
        "access-profile-name": "pppoe",
        "service-profile-name": "nat_service",
        "aaa-profile-name": "ipoe-aaa"
      },
    ]
  }
}
```

#### Enable NAT on an Instance

You can enable address translation for a specific routing instance such as VRF or virtual router rather than globally across the entire router.

**Syntax:**

**set instance** <instance-name> **address-translation true**

| Attribute | Description |
|---|---|
| `<instance-name>` | Name of the instance. |

The following command is used to enable address translation for the instance named 'vrf1'.

```
set instance vrf1 address-translation true
```

```
supervisor@rtbrick.net: op> show config instance vrf1 address-translation
{
  "rtbrick-config:address-translation": "true"
}
```

#### Enable NAT on External Interface

It is required to enable the NAT on the external (core-facing) interface.

**Syntax:**

**set interface** <interface-name> **unit** <unit-id> **address-translation direction** <public>

| Attribute | Description |
|---|---|
| `<interface-name>` | Name of the interface. |
| `<unit-id>` | Configure the number of sub-interfaces under the physical interface. |
| `public` | Specify 'public' for the external interface. |

The following commands configure the external interface `ifp-0/1/64` for IPv4 address translation. 'Unit' logical identifier for this physical interface. Direction 'public' shows the configuration on external interface for address translation.

```
set interface ifp-0/1/64 unit 100
set interface ifp-0/1/64 unit 100 address-translation
set interface ifp-0/1/64 unit 100 address-translation direction public
```

```
supervisor@rtbrick.net: cfg> show config interface ifp-0/1/64 unit 100
{
  "rtbrick-config:unit": [
```

```
    {
      "unit-id": 100,
      "address-translation": {
        "direction": "public"
      }
    }
  ]
}
```

#### Enable Logging for NAT

You can optionally enable logging for CGNAT operations.

> All RBFS logs and related information is available in the *RBFS Logging User Guide*. For the list of RBFS logs, see Log Reference.

**set log bd** <name> <options>

| Attribute | Description |
|---|---|
| `level` | Specify the log level. |
| `module` | Specify the log module. |
| `plugin-alias` | Specify the plugin-alias URL. Plugin-alias is an external logging host server to which you can export logs. For example, Graylog. |

The following commands configure logging for NAT module `natd` with a log level 'debug'.

```
set log bd natd
set log bd natd module nat
set log bd natd module nat level debug
```

```
supervisor@rtbrick.net: cfg> show config log bd natd
{
  "rtbrick-config:bd": [
    {
      "bd-name": "natd",
      "module": [
        {
          "module-name": "nat",
          "level": "debug"
        }
      ]
    }
  ]
}
```

#### Modifying the Policer on NAT Trap

You can modify a 'policer' associated with NAT traps, which directs certain packets to the control plane (hostpath). A NAT trap indicates when a device is performing network address translation, some packets that require special handling, are diverted to the control plane for processing.

To change the policer for NAT traps, use the following command:

**Syntax:**

**set forwarding-options address-translation trap policer** <policer-name>

| Attribute | Description |
|---|---|
| `<policer-name>` | Specify the name of the policer. |

Example for modifying the policer for NAT trap.

```
set forwarding-options address-translation trap policer _DEFAULT_POLICER_5_MB
```

#### Selective NAT Bypass Using Destination Prefix

RBFS CGNAT allows selective bypass of Network Address Translation based on destination IPv4 prefixes using an L3 ACL rule. Traffic matching the configured prefix in the specified direction is excluded from address translation. This feature enables you to preserve original source and destination addresses for specific networks.

**Syntax**

**set forwarding-options acl l3v4 rule** <name> **ordinal** <value> **match destination-ipv4-prefix** <prefix>
**set forwarding-options acl l3v4 rule** <name> **ordinal** <value> **match direction ingress**
**set forwarding-options acl l3v4 rule** <name> **ordinal** <value> **action trap default**
**set forwarding-options acl l3v4 rule** <name> **ordinal** <value> **priority** <value>

The following configuration defines a rule named 'exclude_nat' with an ingress match condition. The rule matches traffic destined to the 10.10.10.1/24 prefix and assigns it a priority of 500 for processing order. Matching packets are trapped to the default control-plane handler, allowing the system to process them separately for excluding them from NAT handling.

```
set forwarding-options acl l3v4 rule exclude_nat
set forwarding-options acl l3v4 rule exclude_nat ordinal 1
set forwarding-options acl l3v4 rule exclude_nat ordinal 1 match destination-ipv4-
prefix 10.10.10.1/24
set forwarding-options acl l3v4 rule exclude_nat ordinal 1 match direction ingress
set forwarding-options acl l3v4 rule exclude_nat ordinal 1 action trap default
```

```
set forwarding-options acl l3v4 rule exclude_nat ordinal 1 priority 500
```

```
{
  "rtbrick-config:acl": {
    "l3v4": {
      "rule": [
        {
          "rule-name": "exclude_nat",
          "ordinal": [
            {
              "ordinal-value": 1,
              "match": {
                "destination-ipv4-prefix": "10.10.10.1/24",
                "direction": "ingress"
              },
              "action": {
                "trap": "default"
              },
              "priority": 500
            }
          ]
        }
      ]
    }
  }
}
```

**Selective NAT Bypass for Subscribers Using ACLs**

For information about match condition, action, prefix list configurations, see the following sections:

- Configuring ACL Match Criteria
- Configuring ACL Actions
- Configuring Prefix Lists

Example:

This configuration defines an ACL rule named 'override_nat' to selectively control NAT behavior for specific subscribers. The rule is assigned ordinal '1' and it matches ingress traffic, meaning packets are evaluated as they enter the device. The match condition uses a prefix list called 'NON_NAT_SUBS', which contains the IPv4 address range 192.0.2.0/24 and 198.51.100.0/24. Any packet with a source IP belonging to these range will match the rule.

The configured action 'trap default' redirects matching packets to the control plane. This overrides the default NAT processing behavior. As a result, traffic from these specified prefix list is excluded from NAT.

The source IPv4 address space must be unique and must not share any address

space with the public NAT pool addresses.

```
set forwarding-options acl l3v4 rule override_nat
set forwarding-options acl l3v4 rule override_nat ordinal 1
set forwarding-options acl l3v4 rule override_nat ordinal 1 match direction ingress
set forwarding-options acl l3v4 rule override_nat ordinal 1 match source-ipv4-prefix-
list NON_NAT_SUBS
set forwarding-options acl l3v4 rule override_nat ordinal 1 action trap default
set forwarding-options prefix-list NON_NAT_SUBS
set forwarding-options prefix-list NON_NAT_SUBS ipv4-prefix 192.0.2.0/24
set forwarding-options prefix-list NON_NAT_SUBS ipv4-prefix 198.51.100.0/24
```

```
"rtbrick-config:forwarding-options": {
      "acl": {
        "l3v4": {
          "rule": [
            {
              "rule-name": "override_nat",
              "ordinal": [
                {
                  "ordinal-value": 1,
                  "match": {
                    "direction": "ingress",
                    "source-ipv4-prefix-list": "NON_NAT_SUBS"
                  },
                  "action": {
                    "trap": "default"
                  }
                }
              ]
            }
          ]
        }
      },
      "prefix-list": [
        {
          "prefix-list-name": "NON_NAT_SUBS",
          "ipv4-prefix": [
            {
              "ipv4-prefix": "192.0.2.0/24"
            },
            {
              "ipv4-prefix": "198.51.100.0/24"
            }
          ]
        }
      ]
    },
```

### 5.3.3. Carrier Grade NAT Operational Commands

#### Carrier Grade NAT Show Commands

The CGNAT operational commands provide detailed information about the address translation operations.

##### NAT Pool Information

This command displays information about IPv4 address pool for a user.

**Syntax:**

**show address-translation allocation** <options>

| Option | Description |
|---|---|
| `pool` | Specify the pool name. |
| `user` | Specify the user name. |

Example for NAT pool allocation.

```
supervisor@rtbrick: op> show address-translation allocation pool
Pool: PUBLIC_NAT_POOL
  Instance: default
  Allocated: 5/4032 (0.12%)
```

Example for allocation details for subscribers

```
supervisor@rtbrick: op> show address-translation allocation user
User                            Original Address     Translated Address   Port Range
ppp-0/1/23/72339069014638594     198.51.100.105       203.0.113.1           1024 -
1055
ppp-0/1/23/72339069014638595     198.51.100.103       203.0.113.2           1024 -
1055
ppp-0/1/23/72339069014638596     198.51.100.101       203.0.113.1           1088 -
1119
ppp-0/1/23/72339069014638597     198.51.100.102       203.0.113.2           1056 -
1087
ppp-0/1/23/72339069014638598     198.51.100.104       203.0.113.1           1056 -
1087
```

##### NAT Rule Details

**Syntax:**

**show address-translation rule** <options>

| Option | Description |
|---|---|
| - | Without any option, the command displays the information for all rules. |
| instance | Displays NAT rule information for the specified instance. |
| summary | Displays summary of the information for the NAT rule. |
| user | Displays user information for the NAT rule. |

Example for address translation rule for an instance.

```
supervisor@rtbrick: op> show address-translation rule instance default
Instance: default
User                              Protocol   Original Address        Translated Address
Direction
ppp-0/1/23/72339069014638596      tcp        198.51.100.101, 60001   203.0.113.1, 1089
Both
ppp-0/1/23/72339069014638597      tcp        198.51.100.102, 60001   203.0.113.2, 1057
Both
ppp-0/1/23/72339069014638595      tcp        198.51.100.103, 60001   203.0.113.2, 1024
Both
ppp-0/1/23/72339069014638598      tcp        198.51.100.104, 60001   203.0.113.1, 1056
Both
ppp-0/1/23/72339069014638594      tcp        198.51.100.105, 60001   203.0.113.1, 1025
Both
ppp-0/1/23/72339069014638596      tcp        198.51.100.101, 60002   203.0.113.1, 1088
Both
ppp-0/1/23/72339069014638597      tcp        198.51.100.102, 60002   203.0.113.2, 1056
Both
ppp-0/1/23/72339069014638595      tcp        198.51.100.103, 60002   203.0.113.2, 1025
Both
ppp-0/1/23/72339069014638598      tcp        198.51.100.104, 60002   203.0.113.1, 1057
Both
ppp-0/1/23/72339069014638594      tcp        198.51.100.105, 60002   203.0.113.1, 1024
Both
ppp-0/1/23/72339069014638596      udp        198.51.100.101, 60001   203.0.113.1, 1089
Both
ppp-0/1/23/72339069014638597      udp        198.51.100.102, 60001   203.0.113.2, 1057
Both
ppp-0/1/23/72339069014638595      udp        198.51.100.103, 60001   203.0.113.2, 1025
Both
ppp-0/1/23/72339069014638598      udp        198.51.100.104, 60001   203.0.113.1, 1056
Both
ppp-0/1/23/72339069014638594      udp        198.51.100.105, 60001   203.0.113.1, 1025
Both
ppp-0/1/23/72339069014638596      udp        198.51.100.101, 60002   203.0.113.1, 1088
Both
```

```
ppp-0/1/23/72339069014638597     udp        198.51.100.102, 60002   203.0.113.2, 1056
Both
ppp-0/1/23/72339069014638595     udp        198.51.100.103, 60002   203.0.113.2, 1024
Both
ppp-0/1/23/72339069014638598     udp        198.51.100.104, 60002   203.0.113.1, 1057
Both
ppp-0/1/23/72339069014638594     udp        198.51.100.105, 60002   203.0.113.1, 1024
Both
ppp-0/1/23/72339069014638596     icmp       198.51.100.101, 256     203.0.113.1, 1088
Both
ppp-0/1/23/72339069014638597     icmp       198.51.100.102, 256     203.0.113.2, 1056
Both
ppp-0/1/23/72339069014638595     icmp       198.51.100.103, 256     203.0.113.2, 1024
Both
ppp-0/1/23/72339069014638598     icmp       198.51.100.104, 256     203.0.113.1, 1056
Both
ppp-0/1/23/72339069014638594     icmp       198.51.100.105, 256     203.0.113.1, 1024
Both
```

Example for address translation rule summary. The example summary shows the total number of NAT rules for TCP flows.

```
supervisor@rtbrick: op> show address-translation rule summary
Instance: default
  Source     Protocol      Rules
  Dynamic    tcp              10
  Dynamic    udp              10
  Dynamic    other             5
  Total Dynamic Rules         25
```

Example for address translation rules for subscribers.

```
supervisor@rtbrick: op> show address-translation rule
Instance: default
User                             Protocol   Original Address        Translated Address
Direction
ppp-0/1/23/72339069014638596     tcp        198.51.100.101, 60001   203.0.113.1, 1089
Both
ppp-0/1/23/72339069014638597     tcp        198.51.100.102, 60001   203.0.113.2, 1057
Both
ppp-0/1/23/72339069014638595     tcp        198.51.100.103, 60001   203.0.113.2, 1024
Both
ppp-0/1/23/72339069014638598     tcp        198.51.100.104, 60001   203.0.113.1, 1056
Both
ppp-0/1/23/72339069014638594     tcp        198.51.100.105, 60001   203.0.113.1, 1025
Both
ppp-0/1/23/72339069014638596     tcp        198.51.100.101, 60002   203.0.113.1, 1088
Both
ppp-0/1/23/72339069014638597     tcp        198.51.100.102, 60002   203.0.113.2, 1056
```

```
Both
ppp-0/1/23/72339069014638595     tcp        198.51.100.103, 60002   203.0.113.2, 1025
Both
ppp-0/1/23/72339069014638598     tcp        198.51.100.104, 60002   203.0.113.1, 1057
Both
ppp-0/1/23/72339069014638594     tcp        198.51.100.105, 60002   203.0.113.1, 1024
Both
ppp-0/1/23/72339069014638596     udp        198.51.100.101, 60001   203.0.113.1, 1089
Both
ppp-0/1/23/72339069014638597     udp        198.51.100.102, 60001   203.0.113.2, 1057
Both
ppp-0/1/23/72339069014638595     udp        198.51.100.103, 60001   203.0.113.2, 1025
Both
ppp-0/1/23/72339069014638598     udp        198.51.100.104, 60001   203.0.113.1, 1056
Both
ppp-0/1/23/72339069014638594     udp        198.51.100.105, 60001   203.0.113.1, 1025
Both
ppp-0/1/23/72339069014638596     udp        198.51.100.101, 60002   203.0.113.1, 1088
Both
ppp-0/1/23/72339069014638597     udp        198.51.100.102, 60002   203.0.113.2, 1056
Both
ppp-0/1/23/72339069014638595     udp        198.51.100.103, 60002   203.0.113.2, 1024
Both
ppp-0/1/23/72339069014638598     udp        198.51.100.104, 60002   203.0.113.1, 1057
Both
ppp-0/1/23/72339069014638594     udp        198.51.100.105, 60002   203.0.113.1, 1024
Both
ppp-0/1/23/72339069014638596     icmp       198.51.100.101, 256     203.0.113.1, 1088
Both
ppp-0/1/23/72339069014638597     icmp       198.51.100.102, 256     203.0.113.2, 1056
Both
ppp-0/1/23/72339069014638595     icmp       198.51.100.103, 256     203.0.113.2, 1024
Both
ppp-0/1/23/72339069014638598     icmp       198.51.100.104, 256     203.0.113.1, 1056
Both
ppp-0/1/23/72339069014638594     icmp       198.51.100.105, 256     203.0.113.1, 1024
Both
```

The example summary shows the total number of NAT rules for other protocol flows:

```
supervisor@rtbrick: op> show address-translation rule
Instance: default
User                             Protocol   Original Address        Translated Address
Direction
ppp-0/1/23/72339069014638596     tcp        198.51.100.101, 60001   203.0.113.1, 1089
Both
ppp-0/1/23/72339069014638597     tcp        198.51.100.102, 60001   203.0.113.2, 1057
Both
ppp-0/1/23/72339069014638595     tcp        198.51.100.103, 60001   203.0.113.2, 1024
Both
```

```
ppp-0/1/23/72339069014638598     tcp        198.51.100.104, 60001   203.0.113.1, 1056
Both
ppp-0/1/23/72339069014638594     tcp        198.51.100.105, 60001   203.0.113.1, 1025
Both
ppp-0/1/23/72339069014638596     tcp        198.51.100.101, 60002   203.0.113.1, 1088
Both
ppp-0/1/23/72339069014638597     tcp        198.51.100.102, 60002   203.0.113.2, 1056
Both
ppp-0/1/23/72339069014638595     tcp        198.51.100.103, 60002   203.0.113.2, 1025
Both
ppp-0/1/23/72339069014638598     tcp        198.51.100.104, 60002   203.0.113.1, 1057
Both
ppp-0/1/23/72339069014638594     tcp        198.51.100.105, 60002   203.0.113.1, 1024
Both
ppp-0/1/23/72339069014638596     udp        198.51.100.101, 60001   203.0.113.1, 1089
Both
ppp-0/1/23/72339069014638597     udp        198.51.100.102, 60001   203.0.113.2, 1057
Both
ppp-0/1/23/72339069014638595     udp        198.51.100.103, 60001   203.0.113.2, 1025
Both
ppp-0/1/23/72339069014638598     udp        198.51.100.104, 60001   203.0.113.1, 1056
Both
ppp-0/1/23/72339069014638594     udp        198.51.100.105, 60001   203.0.113.1, 1025
Both
ppp-0/1/23/72339069014638596     udp        198.51.100.101, 60002   203.0.113.1, 1088
Both
ppp-0/1/23/72339069014638597     udp        198.51.100.102, 60002   203.0.113.2, 1056
Both
ppp-0/1/23/72339069014638595     udp        198.51.100.103, 60002   203.0.113.2, 1024
Both
ppp-0/1/23/72339069014638598     udp        198.51.100.104, 60002   203.0.113.1, 1057
Both
ppp-0/1/23/72339069014638594     udp        198.51.100.105, 60002   203.0.113.1, 1024
Both
ppp-0/1/23/72339069014638596     icmp       198.51.100.101, 256     203.0.113.1, 1088
Both
ppp-0/1/23/72339069014638597     icmp       198.51.100.102, 256     203.0.113.2, 1056
Both
ppp-0/1/23/72339069014638595     icmp       198.51.100.103, 256     203.0.113.2, 1024
Both
ppp-0/1/23/72339069014638598     icmp       198.51.100.104, 256     203.0.113.1, 1056
Both
ppp-0/1/23/72339069014638594     icmp       198.51.100.105, 256     203.0.113.1, 1024
Both
```

**NAT Trap Statistics**

### Syntax:

**show address-translation trap statistics**

This command displays the information of packets trapped in the NAT daemon.

Example for flow stats.

```
supervisor@rtbrick: op> show address-translation trap statistics
Protocol   Flags                     Status             Count
tcp        -|syn|-|-|-|-|-|-|-       Success               10
udp        -|-|-|-|-|-|-|-|-         Success               12
udp        -|-|-|-|-|-|-|-|-         Failure              441
udp_ctrl   -|-|-|-|-|-|-|-|-         Success               64
udp_ctrl   -|-|-|-|-|-|-|-|-         Failure                1
other      -|-|-|-|-|-|-|-|-         Success               86
Total Successful Traps:                                    172
Total Failed Traps:                                        442
```

#### NAT Error Details

This command displays NAT error information.

**Syntax:**

**show address-translation error**

Example for address translation error

```
supervisor@rtbrick: op> show address-translation error
TRAP_HNDL - Fib-Local Route
    Counter        : 494
    Current Update : Fri Jun 05 04:35:33 GMT +0000 2026
    Last Update    : Fri Jun 05 04:35:33 GMT +0000 2026
TRAP_HNDL - DNS Client Port Reused
    Counter        : 6
    Current Update : Fri Jun 05 04:35:26 GMT +0000 2026
    Last Update    : Fri Jun 05 04:35:25 GMT +0000 2026
TRAP_HNDL - Is Duplicate
    Counter        : 1
    Current Update : Fri Jun 05 04:35:23 GMT +0000 2026
    Last Update    : Fri Jun 05 04:35:23 GMT +0000 2026
```

#### Clear NAT Errors

**Syntax**

```
clear address-translation error
```

This command resets the existing statistics for NAT errors.

# 5.4. RBFS Carrier-Grade Network Address Translation Appliance

## 5.4.1. CGNAT Appliance Overview

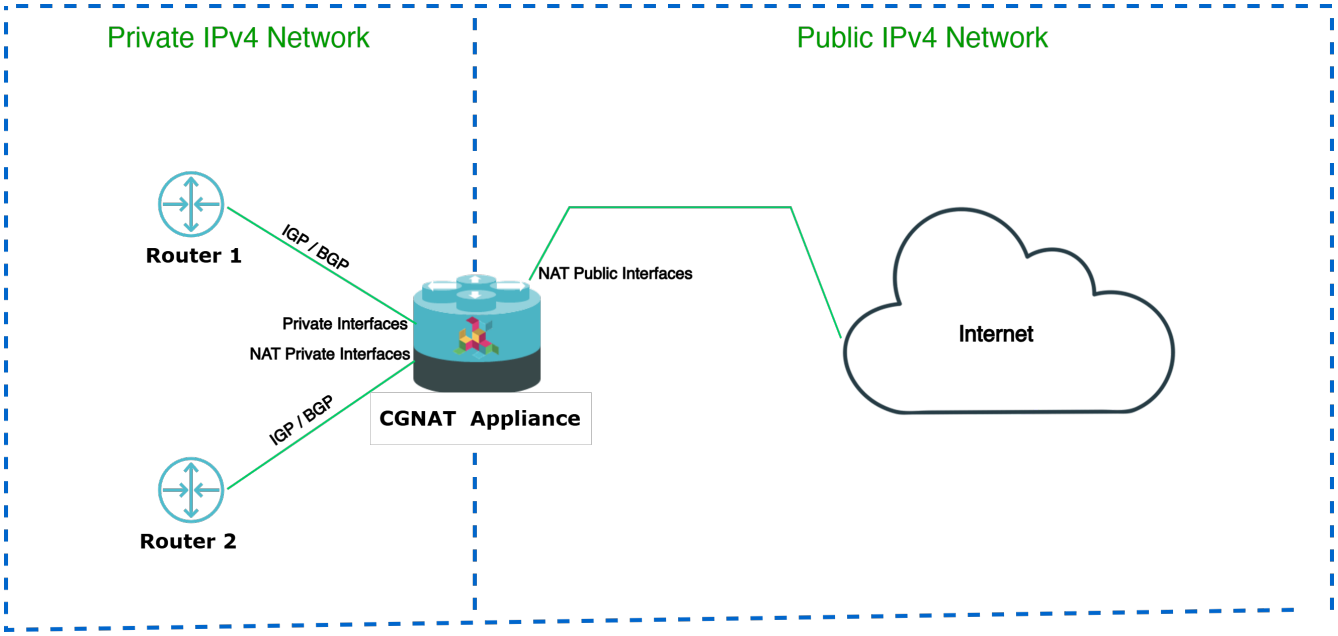
Carrier-Grade Network Address Translation (CGNAT) appliance is a technology solution that addresses the IPv4 exhaustion challenge of service providers. The solution helps to use private network addresses and to limit the use of publicly routable IPv4 addresses significantly.

The main features of the CGNAT appliance are:

- Private Network Routers, such as BNGs, establish reachability within private networks and direct upstream traffic (from the private network to the internet) through a centralized CGNAT appliance. The CGNAT appliance is configured with NAT profiles, public pools, and separate private/public interfaces.
- Downstream reachability is achieved via aggregate NAT routes programmed by the CGNAT appliance and redistributed to BGP peers.
- Upstream reachability involves routers (Router 1 / Router 2 as shown in the diagram below) advertising private pool aggregate routes to the CGNAT appliance.
- Traffic received at CGNAT private interfaces undergoes NAT treatment (SNAT).
- NAT Traffic received at CGNAT public interfaces undergoes NAT treatment(DNAT) and forwarded to the destination, while non-NAT IP traffic is forwarded seamlessly.

The following diagram illustrates a high-level view of CGNAT appliance.

*CGNAT Overview*



There are three basic concepts in address translation:

**Static NAT**

Static NAT associates one private IP address, IP-protocol and source-L4-port with public IP address, IP protocol and source-L4-port. This is a one-to-one mapping and you must manually define the mapping between a private and public address and port (UDP/TCP).

**Dynamic NAT**

In Dynamic NAT, internal private IPv4 addresses and ports are mapped with a public IPv4 address and port, and this happens dynamically. The router dynamically picks an address from the address pool that is not currently assigned.

**NAT444**

CGNAT appliance supports NAT444. NAT 444 refers to three sets of IPv4 addresses: Customer private, ISP private, and public Internet. Network address translation occurs from customer private to ISP private, and then ISP private to public. The NAT444, also known as CGNAT, functionality maps IPv4 subscriber private addresses to IPv4 public addresses.

## Benefits of CGNAT Appliance

CGNAT appliance allows service providers to provide services to a large number of their subscribers using a limited number of public IPv4 addresses. The benefits include:

#### Conserve Public IPv4 Addresses

The solution helps service providers conserve expensive public IPv4 addresses by enabling multiple subscribers to share a single public IPv4 address. With CGNAT, one public IPv4 address can manage hundreds of devices within the private network. It reduces the requirement of continually buying additional public IP addresses.

#### Enhanced Security

In addition to stretching the limited pool of public IPv4 addresses even further, CGNAT also provides significant security benefits. It helps to enhance network security by keeping the internal addressing private from the external network. It makes it difficult for attackers to target specific devices on the network by preventing attacks that target specific IP addresses.

#### Performance at Carrier-Grade

CGNAT appliance solution implements the functions of IPv4 address translation in the network of service providers. Service providers can deploy NAT in a way that allows multiple traffic flows on an L3 interface to share a single global IPv4 address and scale to several thousands of address translations. CGNAT operations are implemented at the forwarding plane (hardware level) to ensure optimal performance. This hardware-level implementation achieves high throughput with no performance impact and packet processing occurs without overburdening the CPU. The solution offers carrier-grade scalability with fast translation rates and a large number of IPv4 address and port number translations.

### Understanding CGNAT Appliance Implementation

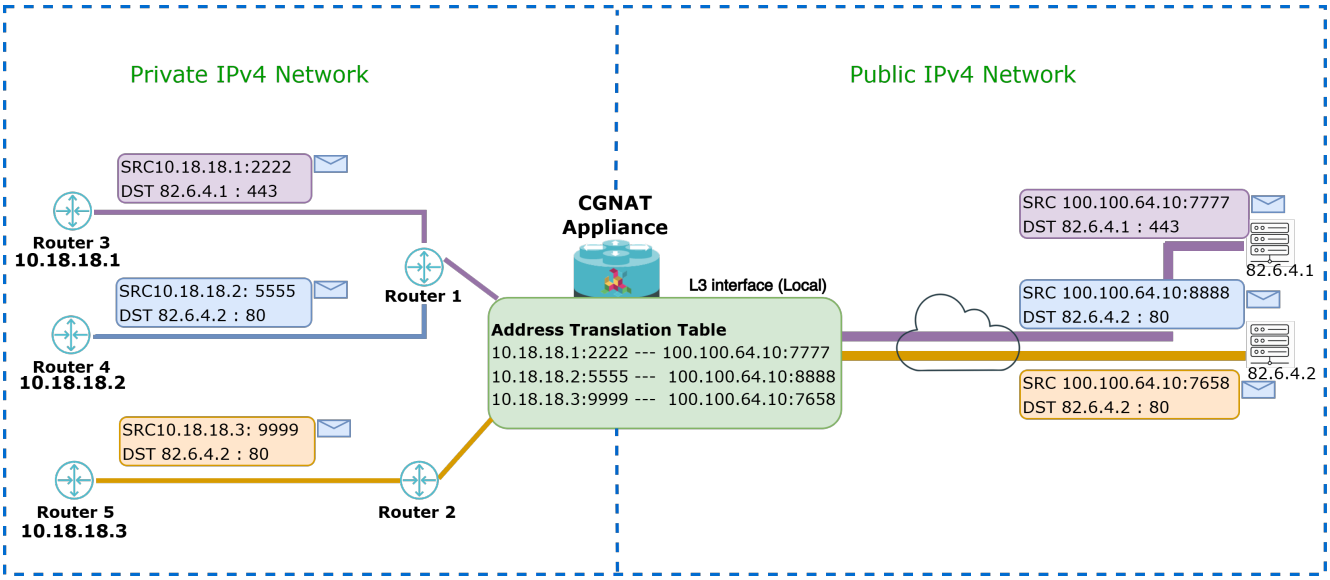
CGNAT appliance solution has been designed to support Port Address Translation, also known as Network Address Port Translation (NAPT), which has the greatest potential to conserve IPv4 addresses for service providers. NAPT is known as an effective method to allow multiple devices to connect to the Internet using a single public IPv4 address.

#### Network Address Port Translation

Network Address Port Translation is a dynamic NAT in which port numbers along with IP address are used to identify which traffic belongs to which private IP address. CGNAT appliance translates the source private IPv4 address and port number to a public source IPv4 address and unique port number. This allows multiple devices with different private IP addresses to use a single public IPv4 address. The unique port number ensures that the traffic is delivered to the correct device.

The following diagrams illustrate how CGNAT appliance works at a high level.

*CGNAT Traffic*




The diagram shows three subscribers using private IPv4 addresses (`10.18.18.1`, `10.18.18.2`, and `10.18.18.3` send traffic to two different servers (`82.6.4.1` and `82.6.4.2`) on the public network.

CGNAT performs the address translation by replacing the source private IPv4 address with the public IPv4 address from the address pool and the source port number with a unique port number. After the address translation, the device forwards the traffic to the destination servers. After the translation, packets have a new source IPv4 address which is the public IPv4 and a unique port number as per the mapping in the address translation table.

CGNAT maintains a translation table with the entries. Entries are records of mapped private IPv4 addresses and port numbers with the public IPv4 addresses and port numbers.

The downstream traffic from the servers traverses to the CGNAT device. The packets coming from external hosts include the destination address as the (translated) public IPv4 address. CGNAT device performs reversal address translation for these packets as per the translation table mapping for the downstream traffic.

NAPT helps to significantly reduce the number of logs generated as it generates logs only during the allocation and release of each block of ports.

#### Deterministic NAT

CGNAT appliance offers support for deterministic NAT mode, which provides a consistent mapping of private IPv4 addresses with public IPv4 addresses and port ranges. This mode ensures a one-to-one mapping of private IPv4 addresses with public IPv4 addresses, allowing you to specify the private address and its matching public address and port range. The given private IPv4 address is always translated to the same public address.

In addition, deterministic NAT guarantees a predetermined and fixed translation for a given internal address with a public IPv4 and port combination, ensuring consistency and stability in the mapping. This feature is particularly useful for applications that involve security protocols, as well as for service providers who need to track private L3 interface traffic flows.

Furthermore, deterministic NAT significantly reduces address translation logs, as private IPv4 addresses are always mapped to public IPv4 addresses and port ranges.

#### Address Translation Table

Port mapping is a feature that allows multiple devices to share the same public IPv4 address with different port numbers. CGNAT generates a unique port number for each traffic flow on the L3 interface. Port mapping helps to determine the correct host among the many devices in the private network that use the same public IPv4 address.

CGNAT appliance maintains an address translation table that maps private IPv4 addresses and port numbers to their corresponding public IPv4 addresses and ports. Whenever an incoming packet arrives, CGNAT checks the translation table to see if a translation entry exists for that packet. If there’s an entry, the CGNAT replaces the source IPv4 address and port number in the packet header with the mapped public IPv4 address and port number.

When a device on the Internet sends packets downstream, the CGNAT software uses the translation table for address translation reversal. It replaces the destination public IPv4 address and port number in the packet header with the corresponding private IPv4 address and port number.

#### NAT IP Pools and Chaining

A NAT pool contains a range of multiple public IP addresses. You can create multiple pools and associate them with a NAT service. Pool chaining is a method in which you can associate one pool with another. For a pool, you can define 'next pool name'; so that when the pool gets exhausted with the IPv4 addresses, the next pool that is defined will take over.

#### Ageing

Ageing refers to the time that a translation entry exists or remains in the address translation table after it was last used.

The entries in the translation table have a finite lifespan as the software implements mechanisms to handle session timeouts.

When a host sends a packet to a destination, CGNAT translates the private IPv4 address and port to a public IPv4 address and port. This mapping is recorded as an entry in the address translation table. The software always looks up the translation table whenever it receives a packet to verify that any entry exists for the packet in the translation table. The software performs the address translation based on the recorded entries and its associated mappings for both inbound and outbound packets.

If there is no activity related to a specific mapping (for example, when there are no incoming or outgoing packets), the mapping will eventually be removed from the address translation table, once idle or unused entries are detected, entries removed from the address translation table to free up resources for the new upcoming traffic flows.

For TCP, idle or unused flows are typically detected by the receipt of TCP FIN-ACK packets, which indicate that both sides of the connection have finished sending data. Once detected, these flows are removed after a configurable aging timer expires.

For UDP, as it is a connectionless protocol, idle or unused flows are detected by periodically polling the flow traffic. As the number of UDP traffic flows increases, the idle flow detection mechanism also increases, ensuring that idle flows are removed in a timely manner. The aging of UDP flows is proportional to the number of traffic flows.

```
NOTE: The NAT ageing timeout is 200 seconds for TCP and UDP traffic. The actual time
taken to flush aged-out NAT rules depends on the number of NAT rules and the number of
counters.
```

#### Logging

CGNAT Appliance provides traffic monitoring capabilities to track and log network activities. You can enable logging for CGNAT operations.

Logging is not completely supported on CGNAT Appliance.

#### CGNAT Handling of ICMP Messages

The NAT-enabled RBFS devices can perform network address translation for ICMP echo request and reply messages. Use the `ping` command (ICMP Echo Request, Echo Reply) to check internet connectivity to external internet addresses when NAT is enabled. This feature is enabled by default.

#### CGNAT Handling of TCP Control Messages

RBFS with NAT supports address translation for TCP control messages using ACLs. It includes the TCP control messages such as FIN, FIN-ACK, RST, RST-ACK, SYN, and SYN-ACK. This feature is enabled by default.

#### CGNAT Support for Traceroute

The RBFS CGNAT provides support for Traceroute and critical ICMP error handling on the Edgecore AGR420, UfiSpace S9600-72XC, and UfiSpace S9600-102XC platforms. This feature ensures that diagnostic tools can function correctly across NAT boundaries by accurately translating and

forwarding "TTL Exceeded" and "Fragmentation Needed" messages.

#### Support for Path MTU Discovery on CGNAT Deployments

Path MTU Discovery works in NAT environments for PPPoE and IPoE subscribers. RBFS with CGNAT-enabled can send ICMP "Fragmentation Needed" messages to the source when packet fragmentation is required, allowing the source to adjust the packet size accordingly. It allows for dynamic MTU adjustment and prevent packet drops and fragmentation related issues.

Limitations:

- NAT for fragmented IPv4 packets that lack TCP or UDP headers is not supported.
- Fragmentation for NAT subscriber traffic is not supported.

#### Guidelines and Limitations

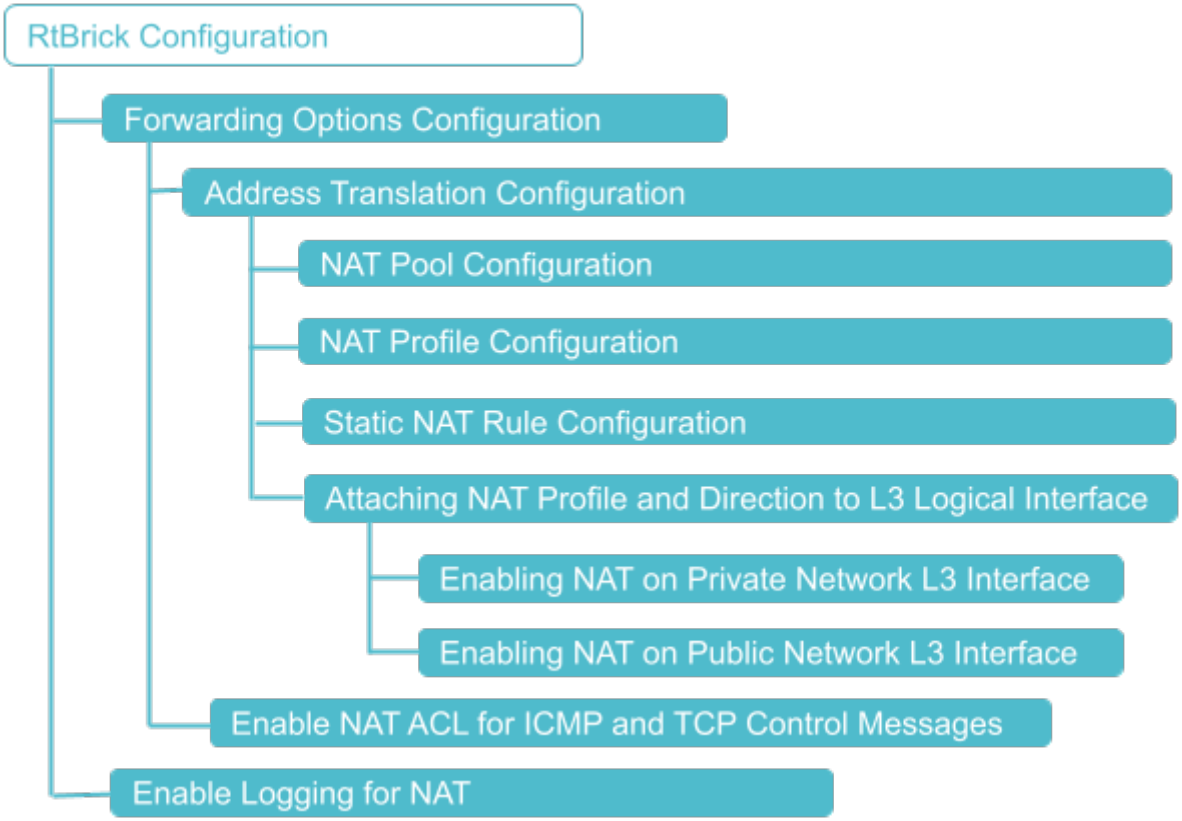
- Before configuring CGNAT, ensure that the correct platform profile is configured. Reboot the system for the changes to take effect, after setting the platform profile. For information about Platform Profile configuration, see Platform Profile.

### Supported Platforms

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

## 5.4.2. CGNAT Appliance Configuration

### Configuration Hierarchy




**Prerequisites**

Before configuring CGNAT, ensure that the appropriate platform profile is configured on the device. The platform profile determines the operating mode of the system and enables the required hardware resources for NAT functionality.

Configure one of the following platform profiles based on the deployment type:

- Use `nat_1q` or `nat_4q` profile when the system operates as a RBFS CGNAT, where NAT is integrated with subscriber management.
- Use this `nat_standalone` profile when the system operates as a dedicated CGNAT appliance without BNG subscriber functionality.

The platform profile must be configured and the system must be rebooted before proceeding with the CGNAT configuration.

Reboot the system for the changes to take effect, after setting the platform profile. For information about Platform Profile configuration, see Platform Profile.

**CGNAT Appliance Configuration**

You must perform the following tasks to configure CGNAT appliance.

1. Configure NAT Pool
2. Configure NAT Profile
3. Enable NAT on an Instance

4. Attaching NAT Profile and Direction to L3 Logical Interface
5. Enable NAT on Private Network L3 Interface
6. Enable NAT on Public Network L3 Interface
7. Configure NAT Rule
8. Enable Logging for NAT

### Configuration Syntax and Commands

The following sections describe the CGNAT appliance configuration syntax and commands.

#### NAT Pool Configuration

A NAT IP address pool includes a set of public IPv4 addresses that are used for network address translation. You can create multiple public IPv4 address pools and one pool includes a range of public IPv4 addresses. These pools allocate public IPv4 addresses to subscribers during address translation. While configuring a pool, you can define the group of public IPv4 addresses belonging to that pool by specifying the lowest and highest IP addresses.

The system allows you to create multiple pools and define the association among them. You can define the 'next-pool-name' that takes over when the current pool gets exhausted with the IPv4 addresses. When one pool gets exhausted, the next pool takes over and starts serving the IP addresses to subscribers when the address translation occurs.

> Static and dynamic IP pools cannot be configured simultaneously; they are mutually exclusive.

**Syntax:**

**set forwarding-options address-translation pool** <pool-name> <attribute> <value>

| Attribute | Description |
|---|---|
| `<pool-name>` | Specify the name of the address pool. |
| `ipv4-address` | Specify both the highest and lowest IPv4 addresses in the range of IPv4 addresses for the pool. |
| `ipv4-address high` | Specify the highest IPv4 address in the address pool. You must specify the highest IP address in the range of IP addresses. |
| `ipv4-address low` | Specify the lowest IPv4 address in the address pool. You must specify the lowest IP address in the range of IP addresses. |
| `next-pool-name` | Specify the name of the next address pool that is to be used when the current address pool is allocated completely. |

Example Configuration:

The following commands configure `nataddr_pool1` as the NAT pool and `nataddr_pool2` as the next

pool.

It indicates NAT pool `nataddr_pool1` contains a range of public IPv4 addresses from `100.100.100.1` to `100.100.100.5`.

When the pool `nataddr_pool1` has fully allocated its IPv4 addresses, the next pool named `nataddr_pool2` will start allocating IPv4 addresses from its pool. The pool `nataddr_pool2` includes a rage of IPv4 address from `100.100.101.1` to `100.100.101.150`.

```
set forwarding-options address-translation pool nataddr_pool1
set forwarding-options address-translation pool nataddr_pool1 next-pool-name
nataddr_pool2
set forwarding-options address-translation pool nataddr_pool1 ipv4-address low
100.100.100.1
set forwarding-options address-translation pool nataddr_pool1 ipv4-address high
100.100.100.5
set forwarding-options address-translation pool nataddr_pool2 ipv4-address low
100.100.101.1
set forwarding-options address-translation pool nataddr_pool2 ipv4-address high
100.100.101.150
```

Example Configuration:

```
supervisor@rtbrick: cfg> show config forwarding-options address-translation pool
{
  "rtbrick-config:pool": [
    {
      "pool-name": "nataddr_pool1",
      "next-pool-name": "nataddr_pool2",
      "ipv4-address": {
        "low": "100.100.100.1",
        "high": "100.100.100.5"
      }
    },
    {
      "pool-name": "nataddr_pool2",
      "ipv4-address": {
        "low": "100.100.101.1",
        "high": "100.100.101.150"
      }
    }
  ]
}
```

**Configuring NAT Profile**

A NAT profile defines how the NAT device has to perform the IPv4 address translation. NAT profile allows you to define an instance, IPv4 address pools, maximum number of translations, and

mapping a particular internal IPv4 address with a particular external IPv4 address for a deterministic address translation.

A single NAT profile can be attached to subscribers across different instances. This means that regardless of which instance a subscriber belongs to, they can share the same NAT profile. For example, subscribers in different routing instances can be managed using the same NAT settings.

Subscribers within the same instance can be assigned different NAT profiles. Different groups of subscribers or services within the same instance can have tailored NAT configurations.

The configuration limit of NAT rules per subscriber in NAT profile is 16,384.

**Syntax:**

**set forwarding-options address-translation profile** <profile-name> <attribute> <value>

| Attribute | Description |
|---|---|
| `<profile-name>` | Specify the NAT profile name. |
| `deterministic [true]` | Specify deterministic as true to enable deterministic NAT for the profile. Deterministic NAT allows subscribers always to connect with a single public IP. |
| `ip-protocol` | Specify the protocol: tcp, udp, or other. |
| `ip-protocol ageing-timeout <ageing-timeout>` | Specify the ageing time value for the protocol. Default, 300 seconds for TCP and UDP. The allowed Ageing Time Out range values are as follows:<br>• TCP: 60 - 86400 seconds<br>• UDP: 120 - 86400 seconds |
| `ip-protocol control-plane-ageing-timeout <ageing-timeout>` | Specify the TCP/UDP control plane ageing timeout value. Allowed range: 5 - 120 seconds.<br><br>* Default, 10 seconds for TCP traffic.<br>* Default, 30 seconds for UDP traffic. |
| `max-rules` | Specify the maximum number of rules per subscriber or an interface. The supported values for maximum address translation for an interface are 64, 128, 256, 512, 1024, 2048, 4096, 8192, and 16384. |
| `pool` | Specify the name of the public IP address pool. |

The following commands configure the NAT profile named `nat_profile1`. The nat profile `nat_profile1` is configured on the instanced `vrf1` with a pool attached `nataddr_pool1`. Maximum rules are configured as 128 rules and the aging period is configured as 600 seconds for TCP traffic, 300 seconds for UDP traffic, and 300 for 'other' traffic.

```
set forwarding-options address-translation profile nat_profile1
set forwarding-options address-translation profile nat_profile1 instance vrf1
set forwarding-options address-translation profile nat_profile1 pool nataddr_pool1
set forwarding-options address-translation profile nat_profile1 ip-protocol tcp
ageing-timeout 600
set forwarding-options address-translation profile nat_profile1 ip-protocol udp
ageing-timeout 300
set forwarding-options address-translation profile nat_profile1 ip-protocol other
ageing-timeout 300
```

Example Configuration:

```
supervisor@rtbrick.net: cfg> show config forwarding-options address-translation
profile
{
  "rtbrick-config:profile": [
    {
      "profile": "nat_profile1",
      "instance": "vrf1",
      "pool": "nataddr_pool1",
      "ip-protocol": {
        "tcp": {
          "ageing-timeout": 600
        },
        "udp": {
          "ageing-timeout": 300
        },
        "other": {
          "ageing-timeout": 300
        }
      }
    }
  ]
}
```

#### Enable NAT on an Instance

You can enable address translation for a specific routing instance such as VRF or virtual router rather than globally across the entire router.

**Syntax:**

**set instance** <instance-name> **address-translation true**

| Attribute | Description |
|---|---|
| `<instance-name>` | Name of the instance. |

The following command is used to enable address translation for the instance named 'vrf1'.

```
set instance vrf1 address-translation true
```

```
supervisor@rtbrick.net: op> show config instance vrf1 address-translation
{
  "rtbrick-config:address-translation": "true"
}
```

#### Attaching NAT Profile and Direction to L3 Logical Interface

##### Enabling NAT on a Private Network L3 Interface

To enable NAT on a private network L3 interface, you need to configure the direction of the network interface as the "local".

**Syntax:**

**set interface** <interface-name> **unit** <unit-id> **address-translation** <attribute> <value>

| Attribute | Description |
|---|---|
| `<interface-name>` | Name of the interface. Example: ifp-0/0/1. |
| `unit <unit-id>` | Create a logical interface (also referred to as a sub-interface) under the physical interface. |
| `profile <profile-name>` | Specify the NAT profile name. |
| `direction local` | Indicates the NAT configuration on a private network. |

The following command attaches the NAT profile `nat_profile` in the `local` direction to an L3 logical interface.

```
supervisor@rtbrick>multiservice-edge1.rtbrick.net: cfg> show config set interface ifp-
0/0/1
set interface ifp-0/0/1
set interface ifp-0/0/1 unit 100
set interface ifp-0/0/1 unit 100 address-translation
set interface ifp-0/0/1 unit 100 address-translation direction local
set interface ifp-0/0/1 unit 100 address-translation profile nat_profile
```

Example Configuration:

```
supervisor@rtbrick>multiservice-edge1.rtbrick.net: cfg> show config  interface ifp-
0/0/1
{
```

```
  "rtbrick-config:interface": [
    {
      "name": "ifp-0/0/1",
      "unit": [
        {
          "unit-id": 100,
          "address-translation": {
            "direction": "local",
            "profile": "nat_profile"
          }
        }
      ]
    }
  ]
}
```

#### Enabling NAT on a Public Network L3 Interface

To enable NAT on a public network L3 interface, you need to configure the direction of the network interface as the "public".

**Syntax:**

**set interface** <interface-name> **unit** <unit-id> **address-translation** <attribute> <value>

| Attribute | Description |
|---|---|
| `<interface-name>` | Name of the interface. |
| `<unit-id>` | Configure the number of sub-interfaces under the physical interface. |
| `direction public` | Specify 'public' for the external interface. |

The following commands configure the external interface `ifp-0/1/64` for IPv4 address translation. 'Unit' logical identifier for this physical interface. Direction 'public' shows the configuration on external interface for address translation.

```
set interface ifp-0/1/64 unit 100
set interface ifp-0/1/64 unit 100 address-translation
set interface ifp-0/1/64 unit 100 address-translation direction public
```

```
supervisor@rtbrick.net: cfg> show config interface ifp-0/1/64 unit 100
{
  "rtbrick-config:unit": [
    {
      "unit-id": 100,
      "address-translation": {
```

```
        "direction": "public"
      }
    }
  ]
}
```

#### NAT Rule Configuration

You can define NAT rules only for static NAT. A NAT rule defines a match condition and a corresponding action. After you specify NAT rules, each packet is matched with each NAT rule. If a packet matches the condition specified in a rule, then the action corresponding to that match occurs. Match rules govern how the translation of private IPv4 addresses to public IPv4 addresses is performed.

With NAT rules, you can define how address translation is applied to traffic, and how to handle various protocols and data traffic, such as TCP and UDP, to ensure proper address translation and the mappings of private addresses to public addresses.

Rules also define how to handle inbound and outbound traffic, different protocols, and data traffic such as TCP and UDP for ensuring the proper address translation of traffic.

A maximum of 16,384 NAT rules on NAT profile is allowed.

**Syntax:**

**set forwarding-options address-translation rule** <rule-name> <attribute> <value>

| Attribute | Description |
|---|---|
| `<rule-name>` | Specify the name of the rule. |
| `ordinal <ordinal-value>` | Specify the ordinal value. An ordinal value is a numerical representation that indicates its relative position or order. |
| `ordinal <ordinal-value> instance` | Specify the RBFS instance name. |
| `ordinal <ordinal-value> ip-protocol [tcp/udp]` | Specify the IP protocol, TCP or UDP. |
| `ordinal <ordinal-value> local [ipv4-address/port]` | Specify the private IPv4 address or port number that needs to be translated. |
| `ordinal <ordinal-value> public [ipv4-address/port]` | Specify the public IPv4 address. This public IP will be mapped with the private IP in the translation table. |

#### Enable Logging for NAT

You can optionally enable logging for CGNAT appliance operations.

All RBFS logs and related information is available in the *RBFS Logging User Guide*.

For the list of RBFS logs, see Log Reference.

**set log bd** <name> <options>

| Attribute | Description |
|---|---|
| `level` | Specify the log level. |
| `module` | Specify the log module. |
| `plugin-alias` | Specify the plugin-alias URL. Plugin-alias is an external logging host server to which you can export logs. For example, Graylog. |

The following commands configure logging for NAT module `natd` with a log level 'debug'.

```
set log bd natd
set log bd natd module nat
set log bd natd module nat level debug
```

```
supervisor@rtbrick.net: cfg> show config log bd natd
{
  "rtbrick-config:bd": [
    {
      "bd-name": "natd",
      "module": [
        {
          "module-name": "nat",
          "level": "debug"
        }
      ]
    }
  ]
}
```

#### Modifying the Policer on NAT Trap

You can modify a 'policer' associated with NAT traps, which directs certain packets to the control plane (hostpath). A NAT trap indicates when a device is performing network address translation, some packets that require special handling, are diverted to the control plane for processing.

To change the policer for NAT traps, use the following command:

**Syntax:**

**set forwarding-options address-translation trap policer** <policer-name>

| Attribute | Description |
| --- | --- |
| `<policer-name>` | Specify the name of the policer. |

Example for modifying the policer for NAT trap.

```
set forwarding-options address-translation trap policer _DEFAULT_POLICER_5_MB
```

#### Selective NAT Bypass Using Destination Prefix

RBFS CGNAT allows selective bypass of Network Address Translation based on destination IPv4 prefixes using an L3 ACL rule. Traffic matching the configured prefix in the specified direction is excluded from address translation. This feature enables you to preserve original source and destination addresses for specific networks.

**Syntax**

**set forwarding-options acl l3v4 rule** <name> **ordinal** <value> **match destination-ipv4-prefix** <prefix> **set forwarding-options acl l3v4 rule** <name> **ordinal** <value> **match direction ingress set forwarding-options acl l3v4 rule** <name> **ordinal** <value> **action trap default set forwarding-options acl l3v4 rule** <name> **ordinal** <value> **priority** <value>

The following configuration defines a rule named 'exclude_nat' with an ingress match condition. The rule matches traffic destined to the 10.10.10.1/24 prefix and assigns it a priority of 500 for processing order. Matching packets are trapped to the default control-plane handler, allowing the system to process them separately for excluding them from NAT handling.

```
set forwarding-options acl l3v4 rule exclude_nat
set forwarding-options acl l3v4 rule exclude_nat ordinal 1
set forwarding-options acl l3v4 rule exclude_nat ordinal 1 match destination-ipv4-
prefix 10.10.10.1/24
set forwarding-options acl l3v4 rule exclude_nat ordinal 1 match direction ingress
set forwarding-options acl l3v4 rule exclude_nat ordinal 1 action trap default
set forwarding-options acl l3v4 rule exclude_nat ordinal 1 priority 500
```

```
{
  "rtbrick-config:acl": {
    "l3v4": {
      "rule": [
        {
          "rule-name": "exclude_nat",
          "ordinal": [
            {
              "ordinal-value": 1,
              "match": {
                "destination-ipv4-prefix": "10.10.10.1/24",
```

```
                "direction": "ingress"
              },
              "action": {
                "trap": "default"
              },
              "priority": 500
            }
          ]
        }
      ]
    }
  }
}
```

### 5.4.3. CGNAT Appliance Operational Commands

#### CGNAT Appliance Show Commands

The CGNAT operational commands provide detailed information about the address translation operations.

##### NAT Pool Information

This command displays information about IPv4 address pool for a user.

**Syntax:**

**show address-translation allocation** <options>

| Option | Description |
|---|---|
| `pool` | Specify the pool name. |
| `user` | Specify the user name. |

Example for NAT pool allocation.

```
supervisor@rtbrick: op> show address-translation allocation pool
Pool: NAT_PUBLIC_POOL
  Instance: default
  Allocated: 4/4032 (0.10%)
```

##### NAT Rule Details

**Syntax:**

**show address-translation rule** <options>

| Option | Description |
| --- | --- |
| - | Without any option, the command displays the information for all rules. |
| instance | Displays NAT rule information for the specified instance. |
| summary | Displays summary of the information for the NAT rule. |
| user | Displays user information for the NAT rule. |

Example for address translation rules for subscribers.

```
supervisor@rtbrick: op> show address-translation rule
Instance: default
User                            Protocol   Original Address        Translated Address
Direction
ifl-0/1/68/0                    tcp        198.51.100.101, 60001   203.0.113.1, 1088
Both
ifl-0/1/68/0                    tcp        198.51.100.102, 60001   203.0.113.2, 1024
Both
ifl-0/1/68/0                    tcp        198.51.100.103, 60001   203.0.113.1, 1025
Both
ifl-0/1/68/0                    tcp        198.51.100.104, 60001   203.0.113.1, 1057
Both
ifl-0/1/68/0                    tcp        198.51.100.105, 60001   203.0.113.2, 1057
Both
ifl-0/1/68/0                    tcp        198.51.100.101, 60002   203.0.113.1, 1089
Both
ifl-0/1/68/0                    tcp        198.51.100.102, 60002   203.0.113.2, 1025
Both
ifl-0/1/68/0                    tcp        198.51.100.103, 60002   203.0.113.1, 1024
Both
ifl-0/1/68/0                    tcp        198.51.100.104, 60002   203.0.113.1, 1056
Both
ifl-0/1/68/0                    tcp        198.51.100.105, 60002   203.0.113.2, 1056
Both
ifl-0/1/68/0                    udp        198.51.100.101, 60001   203.0.113.1, 1089
Both
ifl-0/1/68/0                    udp        198.51.100.102, 60001   203.0.113.2, 1025
Both
ifl-0/1/68/0                    udp        198.51.100.103, 60001   203.0.113.1, 1025
Both
ifl-0/1/68/0                    udp        198.51.100.104, 60001   203.0.113.1, 1056
Both
ifl-0/1/68/0                    udp        198.51.100.105, 60001   203.0.113.2, 1057
Both
ifl-0/1/68/0                    udp        198.51.100.101, 60002   203.0.113.1, 1088
Both
ifl-0/1/68/0                    udp        198.51.100.102, 60002   203.0.113.2, 1024
Both
ifl-0/1/68/0                    udp        198.51.100.103, 60002   203.0.113.1, 1024
```

```
Both
ifl-0/1/68/0                    udp        198.51.100.104, 60002   203.0.113.1, 1057
Both
ifl-0/1/68/0                    udp        198.51.100.105, 60002   203.0.113.2, 1056
Both
```

The example summary shows the total number of NAT rules for other protocol flows:

```
supervisor@rtbrick.net: cfg> show address-translation rule
Instance: vrf1
User                            Protocol   Original Address        Translated Address
Direction
ifl-0/1/6/100                   icmp       10.100.128.1, 7684      100.100.100.100,
1024   Both
```

#### NAT Trap Statistics

**Syntax:**

**show address-translation trap statistics**

This command displays the information of packets trapped in the NAT daemon.

Example for TCP flow stats

```
supervisor@rtbrick: op> show address-translation trap statistics
Protocol   Flags                     Status             Count
tcp        -|syn|-|-|-|-|-|-|-       Success               10
tcp        -|syn|-|-|-|-|-|-|-       Failure                4
udp        -|-|-|-|-|-|-|-|-         Success               10
udp        -|-|-|-|-|-|-|-|-         Failure              204
udp_ctrl   -|-|-|-|-|-|-|-|-         Failure               41
Total Successful Traps:                                     20
Total Failed Traps:                                        249
```

Example for TCP, UDP, ICMP(other) flow stats

```
supervisor@rtbrick: cfg> show address-translation trap statistics
Protocol   Flags                     Status             Count
tcp        -|syn|-|-|-|-|-|-|-       Success               60
tcp        fin|-|-|-|ack|-|-|-|-     Success                4
tcp        -|syn|-|-|ack|-|-|-|-     Failure               12
udp        -|-|-|-|-|-|-|-|-         Success              182
udp        -|-|-|-|-|-|-|-|-         Failure              326
other      -|-|-|-|-|-|-|-|-         Success             5292
Total Successful Traps:                                   5538
```

```
Total Failed Traps:                                    338
```

#### NAT Error Details

This command displays NAT error information.

**Syntax:**

```
show address-translation error
```

Example for address translation error:

```
supervisor@rtbrick: op> show address-translation error
TRAP_HNDL - Nat Profile Not Present
    Counter        : 243
    Current Update : Fri Jun 05 17:15:15 GMT +0000 2026
    Last Update    : Fri Jun 05 17:15:15 GMT +0000 2026
TRAP_HNDL - Fib-Local Route
    Counter        : 50
    Current Update : Fri Jun 05 17:15:10 GMT +0000 2026
    Last Update    : Fri Jun 05 17:15:10 GMT +0000 2026
TRAP_HNDL - Source route requested
    Counter        : 50
    Current Update : Fri Jun 05 17:15:10 GMT +0000 2026
    Last Update    : Fri Jun 05 17:15:10 GMT +0000 2026
```

#### Clear NAT Errors

**Syntax:**

```
clear address-translation error
```

This command resets the existing statistics for NAT errors.

# 5.5. RBFS HTTP Redirect Service

## 5.5.1. RBFS HTTP Redirect Service Overview

This document provides information about RBFS HTTP Redirect Service. For enabling HTTP Redirect service, it is required to create, associate, and apply subscriber filters (ACLs) for the subscribers. For information about subscriber filters and how to configure these filters, refer to the `RBFS Subscriber Filters User Guide`.

RBFS HTTP Redirect supports both IPoE and PPPoE subscribers.

RBFS HTTP Redirect service allows network service providers to intercept and redirect HTTP

request traffic from subscribers to a designated captive portal instead of the original destination. This service is useful in a multitude of use cases, ranging from subscriber re-authentication to enforcing acceptance of network usage policies.

This captive portal is a webpage where the redirected subscribers are landed up to fulfill certain actions or conditions before they are granted broader access to the network resources. There are various reasons why captive portals can be set up, such as the following:

- Accept the terms of service.
- Receive and manage HTTP requests to unauthorized web resources.
- Present a web page that requires the completion of certain actions from the subscriber.
- Serve commercial communication or network usage policy messages.

By implementing the RBFS HTTP Redirect Service, network service providers can efficiently manage user access and enforce compliance with network regulations and policies, ultimately enhancing the overall security and user experience within their network environment.

Based on the applied filters, RBFS performs three actions which are `accept`, `drop`, and `redirect`. The action `Accept` allows the subscribers to access the network resource that they request. The `Drop` action restricts the subscriber from accessing the network resource. Finally, the `Redirect` action, if enabled HTTP redirect service, takes the subscriber to a different portal where the service provider wants to fulfill certain actions by the subscriber before accessing the network resource.

In addition to RBFS Subscriber Filters, a redirect action is supported by RBFS through the utilization of Ascend Data Filters (ADF), as described in the RBFS RADIUS Services Guide.

The RBFS HTTP Redirect Service together with the Subscriber Filters empowers network service providers to intercept and redirect HTTP request traffic from subscribers, guiding it towards a designated captive portal instead of its original destination.

#### How RBFS HTTP Redirect Service Works

HTTP requests from subscribers to any destination are intercepted by the RBFS HTTP Redirect service, if the subscriber filter rules are applied to the subscriber. In response to the request from the subscriber to access the network, the HTTP Redirect service provides the HTTP status code 302 along with the new URL to guide the subscriber to the new destination. To make it possible, the RBFS Subscriber Filters are employed, enabling the service to decide which requests should be redirected and which requests should be passed directly without redirection. In addition to RBFS Subscriber Filters, a redirect action is supported by RBFS through the utilization of Ascend Data Filters (ADF), as described in the RBFS RADIUS Services Guide.

#### Enabling HTTP Redirect Service Using RADIUS ADF

Both the RBFS Subscriber Filters and the HTTP Redirect Service can be dynamically enabled, disabled, and updated through RADIUS `access-accept` and `CoA` requests without requiring the re-establishment of the subscriber session. This flexibility allows network administrators to efficiently manage and modify these services as needed without disrupting subscriber connectivity.

RBFS provides a set of vendor-specific attributes to control the RBFS Subscriber Filters and the

HTTP Redirect Service. The attributes include:

- VSA 26-50058-75 - RtBrick-HTTP-Redirect-URL
- VSA 26-50058-76 - RtBrick-IPv4-ACL-IN
- VSA 26-50058-77 - RtBrick-IPv4-ACL-OUT
- VSA 26-50058-78 - RtBrick-IPv6-ACL-IN
- VSA 26-50058-79 - RtBrick-IPv6-ACL-OUT

For more information about these attributes, see RBFS RADIUS Services Guide.

**Limitations**

The HTTP Redirect Service is supported for IPoE and PPPoE subscribers. HTTP Redirect service can redirect only HTTP traffic. Typically, the majority of web requests are in HTTPS format, which cannot be redirected. Due to this limitation, most end-user devices, including PCs, smartphones, and tablets, automatically attempt to access specific well-known URLs to search for captive portals immediately after establishing a network connection. Two common examples are http://connectivitycheck.gstatic.com (vendor-independent) and http://captive.apple.com (Apple devices).

## 5.5.2. HTTP Redirect Service Configuration

The configuration hierarchy for the HTTP Redirect service is illustrated in the diagram.

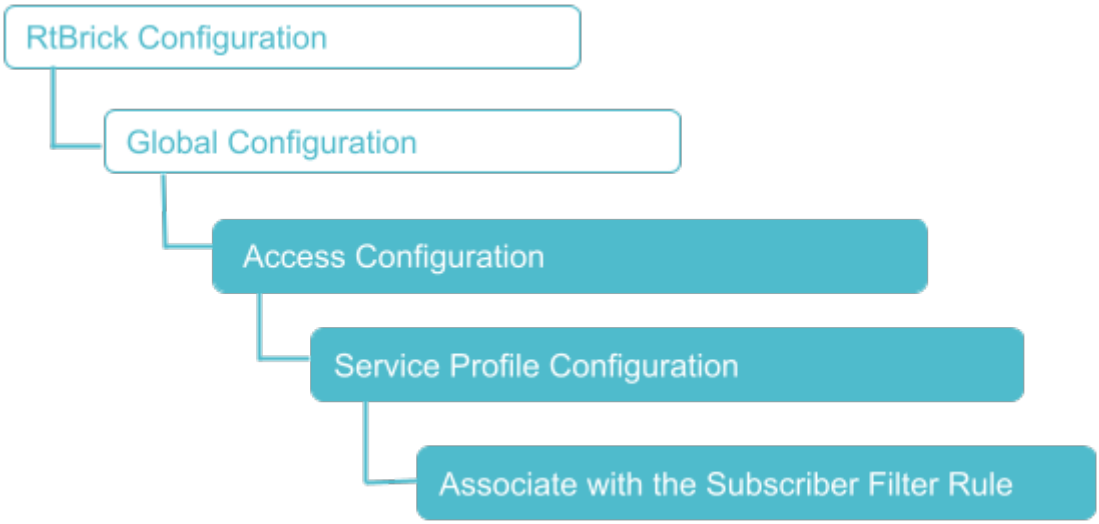



*Figure 88. HTTP Redirect Configuration Hierarchy*

**Service Profile Configuration**

Syntax:

```
set access service-profile profile-name <profile-name> <attribute> <value>
```

| Attribute | Description |
|---|---|
| `<profile-name>` | Service profile name. |
| `<http-redirect>` | HTTP redirect service configuration. |
| `<url>` | HTTP redirect target URL. |
| `<acl>` | Subscriber ACL (filter) configuration. |
| `<ipv4-acl-in>` | IPv4 upstream ACL (ingress from subscriber). |
| `<ipv4-acl-out>` | IPv4 downstream ACL (egress to subscriber). |
| `<ipv6-acl-in>` | IPv6 upstream ACL (ingress from subscriber). |
| `<ipv6-acl-out>` | IPv6 downstream ACL (egress to subscriber). |

Example:

The following configuration specifies the HTTP redirect URL as: https://www.rtbrick.com. It associates the subscriber ACL with the profile. The ACL named 'redirect-acl-in' will determine which traffic is matched and redirected.

```
set access service-profile HTTP
set access service-profile HTTP http-redirect url https://www.rtbrick.com
set access service-profile HTTP acl ipv4-acl-in redirect-acl-in
```

The following example shows the HTTP redirect URL for the service profile.

```
supervisor@router: cfg> show config access service-profile HTTP
{
  "rtbrick-config:service-profile": [
    {
      "profile-name": "HTTP",
      "http-redirect": {
        "url": "https://www.rtbrick.com"
      },
      "acl": {
        "ipv4-acl-in": "redirect-acl-in"
      }
    }
  ]
}
```

### 5.5.3. HTTP Redirect Operational Commands

#### HTTP Redirect Show Commands

The show commands provide detailed information about the HTTP Redirect services.

#### HTTP Redirect Information

You can use the operational commands for monitoring and managing HTTP redirect services. These commands enable you to view configured HTTP redirect services, display operational statistics, and clear collected statistics.

**Syntax:**

**show http-redirect** <option>

| Attribute | Description |
|---|---|
| `statistics` | Displays operational statistics for HTTP redirect services for monitoring. |
| `services` | Displays the configured HTTP redirect services on the device. |

Example:

```
show http-redirect services
show http-redirect statistics
```

Example output for HTTP redirect statistics.

```
supervisor@rtbrick.net: cfg> show http-redirect statistics
HTTP Response Sent:
302 Redirect    : 4
404 Not-Found   : 0

Errors:
IFL Missing     : 0
Context Missing : 0
Duplicate TCP   : 0

TCP Events:
Accepted        : 4
Received        : 4
Sent            : 4
Pollout         : 0
Error           : 0
Close           : 4
Closed          : 0
```

#### Clear Commands

The `clear http-redirect statistics` is used to reset the HTTP redirect statistics counters.

**Syntax**

**clear oam bfd session** <option>

Example command:

```
clear http-redirect statistics
```

# 5.6. RBFS Subscriber Filters

## 5.6.1. RBFS Subscriber Filters Overview

RBFS Subscriber Filters, also referred to as subscriber ACLs, consist of a set of rules defining packet match criteria and actions. There are separate rules for IPv4 and IPv6 downstream (egress to subscriber) and upstream (ingress from subscriber) packets. These rules support various match criteria and actions, some of which are specific to address families or directions. Each rule is assigned a priority, and the decision between multiple matching rules is based on these priorities, where lower values take precedence.

The available actions include `accept`, `drop`, or `http-redirect` where the last one refers to the RBFS HTTP Redirect Service. When the action is `drop`, matching traffic is silently discarded. The filters are categorized into two primary types, namely `l3v4` for IPv4 and `l3v6` for IPv6, applicable to either ingress or egress direction.

To apply these filters to subscribers, there are two ways. They can be applied through the access `service-profile` or directly using the corresponding RADIUS attributes with the second method taking priority.

### About the Match Criteria

When multiple match criteria are defined within a single rule, they are treated as a logical `AND` operation, requiring all criteria to be met for the rule to be considered as a match. However, using unsupported match criteria, such as `destination-ipv4-subscriber-prefix` in ingress (upstream), can potentially lead to session termination. In the case of `CoA` (Change of Authorization), the filter assignment is rejected using `CoA NAK`, if such unsupported criteria are encountered.

Even if filters are assigned to a subscriber, those filters are applied globally, indicating that all traffic from all interfaces and subscribers is evaluated against all rules. Consequently, RBFS has introduced specific options to restrict rules to individual subscribers. For ingress (upstream) rules, it is recommended to enable the `subscriber-ifl` option, ensuring that only traffic received from the corresponding subscriber is matched. With the `subscriber-ifl` option, packets are matched based on incoming subscriber IFL. However, this option is not supported in egress(downstream), requiring the limitation of traffic using subscriber address prefix information. Thus, RBFS introduced the options `source-ipv4-subscriber-prefix`, `source-ipv6-subscriber-prefix`, `destination-ipv4-subscriber-prefix`, `destination-ipv6-subscriber-prefix`, `source-ipv6-delegated-subscriber-prefix`, and `destination-ipv6-delegated-subscriber-prefix`. With these options enabled, the dynamically assigned subscriber address prefix is automatically integrated into the corresponding filter instance to constrain those rules to a specific subscriber.

Improperly configured filters assigned to one subscriber may create a negative impact on other subscribers as well.

### 5.6.2. Subscriber Filters Configuration

The configuration hierarchy for Subscriber ACL is illustrated in the diagram.

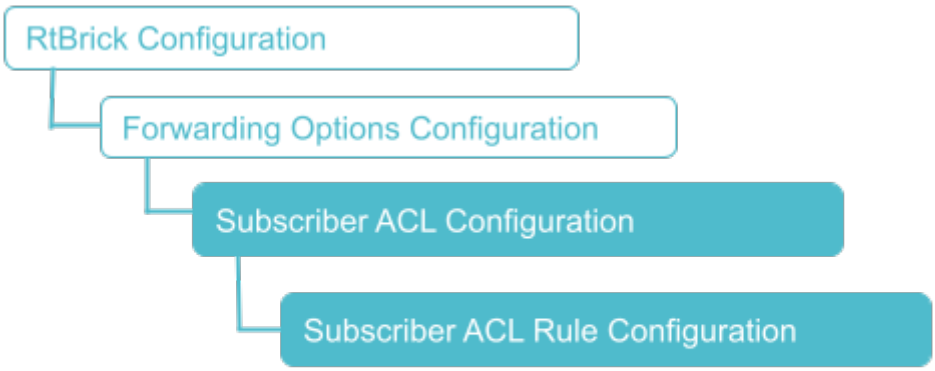



*Figure 89. Subscriber Filters Configuration Hierarchy*

**Configure Subscriber Filters**

**Syntax:**

**set forwarding-options subscriber-acl** <l3v4|l3v6> **rule** <rule-name> **ordinal** <ordinal-value> <option> <attribute> <value>

| Attribute | Description |
|---|---|
| `<l3v4 \| l3v6>` | Specify `l3v4` for IPv4 and `l3v6` for IPv6. |
| `<rule-name>` | Subscriber ACL rule name. |
| `<ordinal-value>` | The mandatory ordinal value is used to differentiate multiple rules within the rule set. |
| `action` | The desired action, which can be either `permit`, `drop` or `http-redirect`. |
| `priority <priority-value>` | The priority of the rule, where the lower value has a higher priority. |
| `match` | The match criteria. |

**Configuring Subscriber Filter Match Criteria**

**Syntax:**

**set forwarding-options subscriber-acl** <l3v4|l3v6> **rule** <rule-name> **ordinal** <ordinal-value> **match** <attribute> <value>

| Attribute | Description |
|---|---|
| `destination-ipv4-prefix <destination-ipv4-prefix>` | Packets are matched when the IPv4 destination address is within the defined prefix. |
| `destination-ipv4-prefix-list <destination-ipv4-prefix-list>` | Packets are matched when the IPv4 destination address is within one of the prefixes listed in the defined prefix list. |
| `destination-ipv4-subscriber-prefix true` | Packets are matched when the IPv4 destination address is within the dynamically assigned subscriber IPv4 address prefix (sometimes, referred to framed prefix). Consequently, this option shares similarity to the `destination-ipv4-prefix`, where the prefix is automatically set to the dynamic subscriber prefix. This option is allowed only in the egress (downstream) direction. |
| `destination-ipv6-prefix <destination-ipv6-prefix>` | Packets are matched when the IPv6 destination address is within the defined prefix. |
| `destination-ipv6-prefix-list <destination-ipv6-prefix-list>` | Packets are matched when the IPv6 destination address is within one of the prefixes listed in the defined prefix list. |
| `destination-ipv6-subscriber-prefix true` | Packets are matched when the IPv6 destination address is within the dynamically assigned subscriber IPv6 address prefix (sometimes, referred to framed prefix). Consequently, this option shares similarity to the destination-ipv4-prefix, where the prefix is automatically set to the dynamic subscriber prefix. This option is allowed only in the egress (downstream) direction. |
| `destination-ipv6-delegated-subscriber-prefix <destination-ipv6-delegated-subscriber-prefix>` | This option is similar to the `destination-ipv6-subscriber-prefix` using the dynamically delegated prefix instead. |
| `source-ipv4-prefix <source-ipv4-prefix>` | Packets are matched when the IPv4 source address is within the defined prefix. |
| `source-ipv4-prefix-list <source-ipv4-prefix-list>` | Packets are matched when the IPv4 source address is within one of the prefixes listed in the defined prefix list. |
| `source-ipv4-subscriber-prefix <source-ipv4-subscriber-prefix>` | Packets are matched when the IPv4 source address is within the dynamically assigned subscriber IPv4 address prefix (sometimes, referred to framed prefix). Consequently, this option shares similarity to `source-ipv4-prefix`, where the prefix is automatically set to the dynamic subscriber prefix. This option is allowed only in the ingress (upstream) direction. |
| `source-ipv6-prefix <source-ipv6-prefix>` | Packets are matched when the IPv6 source address is within the defined prefix. |
| `source-ipv6-prefix-list <source-ipv6-prefix-list>` | Packets are matched when the IPv6 source address is within one of the prefixes listed in the defined prefix list. |

| Attribute | Description |
|---|---|
| `source-ipv6-subscriber-prefix <source-ipv6-subscriber-prefix>` | Packets are matched when the IPv6 source address is within the dynamically assigned subscriber IPv6 address prefix (sometimes, referred to framed prefix). Consequently, this option shares similarity to `source-ipv4-prefix`, where the prefix is automatically set to the dynamic subscriber prefix. This option is allowed only in the ingress (upstream) direction only. |
| `source-ipv6-delegated-subscriber-prefix true` | This option is similar to the `source-ipv6-subscriber-prefix` using the dynamically delegated prefix instead. To disable this configuration, use the `delete` form of the command. |
| `source-l4-port <source-l4-port>` | Packets are matched based on the layer 4 source port (TCP or UDP port). |
| `destination-l4-port <destination-l4-port>` | Packets are matched based on the layer 4 destination port (TCP or UDP port). |
| `ip-protocol <ip-protocol>` | Packets are matched depending on the IP protocol (TCP or UDP). However, this option is not compatible with the MPLS-encapsulated traffic. Consequently, filtering based on IP protocol is not possible for MPLS traffic received from the core. For instance, it is feasible to drop all traffic to port 80, but it is not possible to selectively drop only TCP traffic while permitting UDP traffic when receiving traffic with an MPLS label. |
| `subscriber-ifl true` | Packets are matched based on incoming subscriber IFL. This option is allowed only in the ingress (upstream) direction. |

#### Configuring Subscriber Filter Actions

**Syntax:**

> **set forwarding-options subscriber-acl** <l3v4|l3v6> **rule** <rule-name> **ordinal** <ordinal-value> **action** <attribute> <value>

| Attribute | Description |
|---|---|
| `permit` | Forward packets. |
| `drop` | Silently discard packets. |
| `http-redirect true` | Specify true to enable the redirect service. To disable http-redirect, use the `delete` form of the command. |

Example:

The following configuration sets an ACL for subscriber traffic that targets TCP traffic directed to port 80 (HTTP). The rule is named `ipv4-acl-in`. The IP protocol used is TCP, and the transport port is set to 80 (indicating HTTP traffic). When the specified conditions are met, the action is to redirect traffic to a captive portal. The configuration also sets the priority of the rule. The rule is assigned a

specific priority value, that is 1001. Higher numeric value takes a lower priority.

```
set forwarding-options subscriber-acl l3v4 rule ipv4-acl-in
set forwarding-options subscriber-acl l3v4 rule ipv4-acl-in ordinal 1
set forwarding-options subscriber-acl l3v4 rule ipv4-acl-in ordinal 1 match
destination-l4-port 80
set forwarding-options subscriber-acl l3v4 rule ipv4-acl-in ordinal 1 match ip-
protocol tcp
set forwarding-options subscriber-acl l3v4 rule ipv4-acl-in ordinal 1 action http-
redirect true
set forwarding-options subscriber-acl l3v4 rule ipv4-acl-in ordinal 1 priority 1001
supervisor@M3-STD-44-4403>bm15-tst.fsn.rtbrick.net: cfg> show config  forwarding-
options subscriber-acl
```

```
{
  "rtbrick-config:subscriber-acl": {
    "l3v4": {
      "rule": [
        {
          "rule-name": "ipv4-acl-in",
          "ordinal": [
            {
              "ordinal-value": 1,
              "match": {
                "destination-l4-port": 80,
                "ip-protocol": "tcp"
              },
              "action": {
                "http-redirect": "true"
              },
              "priority": 1001
            }
          ]
        }
      ]
    }
  }
}
```

#### Attaching the ACL Rule with the Service Profile

**Syntax:**

**set access service-profile profile-name** <profile-name> <attribute> <value>

| Attribute | Description |
|---|---|
| `<profile-name>` | Service profile name. |
| `<http-redirect>` | HTTP redirect service configuration. |
| `<url>` | HTTP redirect target URL. |
| `<acl>` | Subscriber ACL (filter) configuration. |
| `<ipv4-acl-in>` | IPv4 upstream ACL (ingress from subscriber). |
| `<ipv4-acl-out>` | IPv4 downstream ACL (egress to subscriber). |
| `<ipv6-acl-in>` | IPv6 upstream ACL (ingress from subscriber). |
| `<ipv6-acl-out>` | IPv6 downstream ACL (egress to subscriber). |

## 5.6.3. Subscriber Filters Operational Commands

### Subscriber Filters Show Commands

The show commands provide detailed information about the subscriber filter.

#### Subscriber ACL (Filter) Information

The `show subscriber <id> acl` command provides a comprehensive list of all ACL instances initiated for the subscriber, including RBFS Subscriber Filters and Ascend Data Filters (ADF). Additionally, this command provides the option to view detailed information for each filter instance by appending the corresponding filter name. Consequently, the filters are displayed with all variables, such as `destination-ipv4-subscriber-prefix`, replaced by their actual prefixes for clarity.

**Syntax:**

**show subscriber** <subscriber-id> **acl** <acl_name> **type (l3v4 | l3v6)**

| Attribute | Description |
|---|---|
| `<subscriber-id>` | Identifier of the subscriber for which to display ACL information. |
| `<acl_name>` | Name of the ACL associated with the subscriber. |
| `type (l3v4\|l3v6)` | Type of the ACL, either IPv4 (l3v4) or IPv6 (l3v6). |

Example: ACL summary information for a specific subscriber

```
supervisor@rtbrick: op> show subscriber 72339069014638594 acl
Rule                                                Ordinal    Priority     Type
REDIRECT_SUBS-ppp-0/1/23/72339069014638594          1          1000         l3v4
                                                    2          1000         l3v4
```

Example: Summary information for a specific subscriber ACL

```
supervisor@rtbrick: op> show subscriber 1369375761697341441 acl ipv4-acl-in-ipoe-lag-
1/1369375761697341441
Rule: ipv4-acl-in-ipoe-lag-1/1369375761697341441
  ACL type: l3v4
  Ordinal: 1
  Priority: 1001
    Match:
      Direction: ingress
      Destination L4 port: 80
      IP protocol: TCP
    Action:
      HTTP-redirect: True       URL: www.rtbrick.com
```

Example: Display the ACL information for a subscriber '72339069014638594' for a specified ACL.

```
supervisor@rtbrick: op> show subscriber 72339069014638594 acl REDIRECT_SUBS-ppp-
0/1/23/72339069014638594
Rule: REDIRECT_SUBS-ppp-0/1/23/72339069014638594
  ACL type: l3v4
  Ordinal: 1             Priority: 1000
    Match:
      Attachment point: ppp-0/1/23/72339069014638594
      Direction: ingress
      Source IPv4 prefix: 198.51.100.101/32
      IP protocol: tcp
    Action:
      HTTP redirect: True   URL: None
  Ordinal: 2             Priority: 1000
    Match:
      Attachment point: ppp-0/1/23/72339069014638594
      Direction: ingress
      Source IPv4 prefix: 198.51.100.102/32
      IP protocol: tcp
    Action:
      HTTP redirect: True   URL: None
```

# 5.7. Lawful Interception

## 5.7.1. Lawful Interception Overview

Lawful Interception (LI) is a legal requirement mandated by national regulations in many regions. It enables authorized Law Enforcement Agencies (LEA) to access specific communication data from Communication Service Providers (CSP).

In RBFS, Lawful Interception enables the selective capture of subscriber sessions in both directions (upstream and downstream). The intercepted traffic is tunneled to a Mediation Device (MD), along with metadata indicating the direction of capture and a reference to the intercepted connection or

session.

RBFS supports LI in accordance with applicable technical standards and regulatory requirements. The system enables dynamic rule provisioning and role-based access control, ensuring that only authorized personnel can configure or view LI-related information. All interception activities can be logged for auditing and compliance purposes.

**Components of Lawful Interception**

The figure below shows the different components of the LI solution.

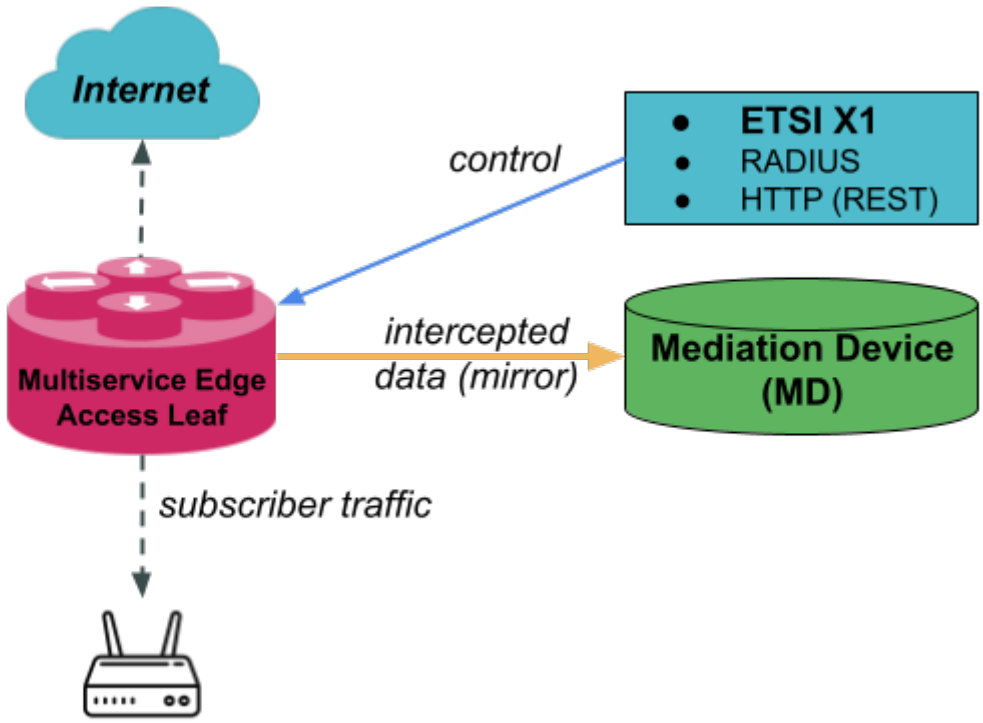



*Figure 90. Components of Lawful Interception*

RBFS supports multiple integration methods to implement Lawful Interception (LI), providing flexibility to match the operator's legal and technical requirements. The following three interfaces are supported for provisioning and managing interception sessions:

1. ETSI X1 Interface (see ETSI TS 101 671)
2. RADIUS Access-Accept and CoA using encrypted attributes
3. HTTP (REST) API

All three methods allow RBFS to set up interception sessions with precise control over subscriber identification, traffic direction, and metadata enrichment. Intercepted traffic is then tunneled to a Mediation Device (MD) using protocols compliant with national regulatory requirements. Role-based access control and full logging ensure that all LI-related actions are secure, traceable, and compliant.

**Packet Format Encoding for LI Headers**

Packet Format Encoding describes the structure of the Lawful Interception (LI) header, which is a

32-bit construct.. This comprises four distinct fields: 3 bits for Direction, 4 bits for Packet type, 3 bits for Sub-Packet type, and 22 bits for LI ID. Each field serves a specific purpose in data transmission.

- **Direction (3 bits)**: Indicates the flow of data.
- **Packet type (4 bits)**: Specifies the kind of packet being transmitted.
- **Sub-Packet type (3 bits)**: Provides a more granular classification within a given packet type.
- **LI ID (22 bits)**: An identifier for the LI header itself.

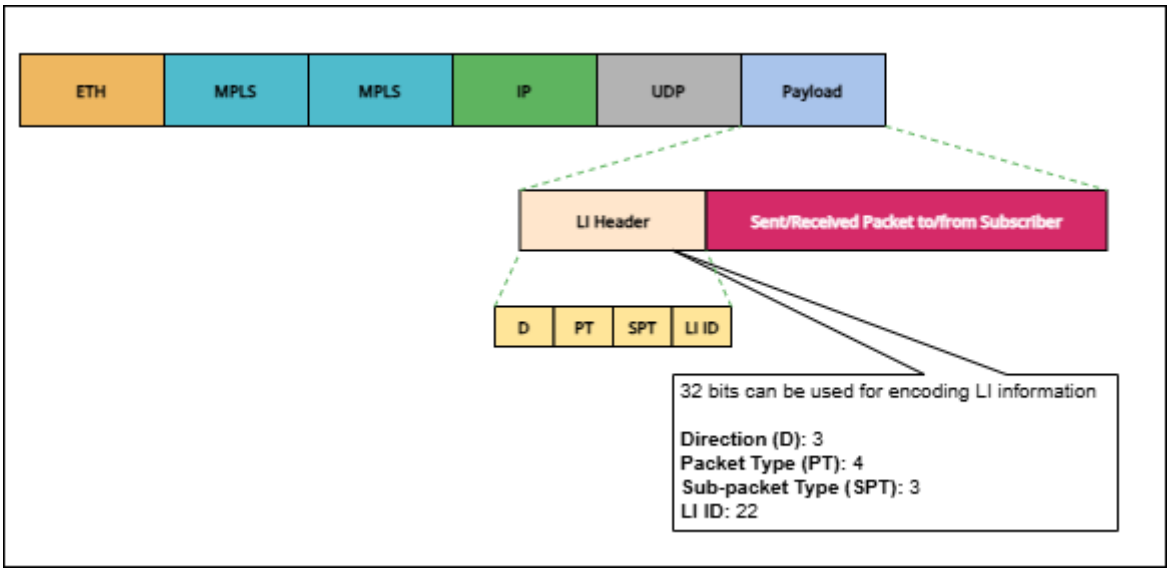



*Figure 91. Packet Format Encoding*

**Payload Direction:**

Payload Direction indicates the flow of data and has the following values:

| Value | Payload Direction |
|---|---|
| 1 | Reserved for keepalive mechanism |
| 2 | Intercepted data or event was sent to target (downstream) |
| 3 | Intercepted data or event was sent from target (upstream) |

**Mapping Payload Format:**

Mapping Payload Format has the following values:

| Value | Payload Format |
|---|---|
| 0-3 | Reserved (unused) |
| 4 | Unknown, Not able to decide the PT |
| 5 | IPv4 packet (not used) |
| 6 | IPv6 packet (not used) |
| 7 | Ethernet Frame (used for Lawful Interception) |

**Sub-payload Format (Type):**

The sub-payload formats are used in the context of Lawful Interception (LI) to classify and manage intercepted data. These formats help in providing a more granular classification within a given packet type.

The sub-payload formats are:

- Single VLAN tag
- Double VLAN tag
- Untagged

**Guidelines and Limitations**

- When LI is enabled on a Layer 3 logical interface (L3IFL), MPLS packets that arrive on this L3IFL will not be mirrored; however, IP packets arriving on the L3IFL will be successfully mirrored.

**Supported Platforms**

Feature availability may vary between hardware platforms. Refer to the Platform Guide for detailed information on which features and sub-features are supported on each specific platform.

### 5.7.2. ETSI X1

Lawful Interception (LI) involves defined interfaces between Law Enforcement Agencies (LEAs) and Communication Service Providers (CSPs), commonly referred to as handover interfaces. Within the CSP domain, LI defines the following three interfaces:

- **X1** for interception management and administrative control
- **X2** for delivery of intercept-related information (IRI)
- **X3** for delivery of intercepted content (CC)

RBFS implements the X1 interface in compliance with **ETSI TS 101 671**. This interface is used for provisioning, modifying, and terminating interception sessions and forms the control plane between the LI management system and RBFS.

Refer to the “LIX1 Configuration” section below for details on configuring the LIX1 administration interface.

**LIX1 Configuration**

In this section, you will find the configurations related to the LIX1 interface.

**Syntax**:

```
set lawful-intercept protocol <protocol-name> <attribute> <value>
```

| Attribute | Description |
| --- | --- |
| `<protocol-name>` | Enables lawful intercept on the switch. Supported Value: x1. |
| `administrative-function-endpoint <administrative-function-endpoint>` | Specifies the ADMF endpoint URL for sending requests. |
| `administrative-function-id <administrative-function-id>` | Specifies the ADMF identifier. |
| `mediation-device-instance <mediation-device-instance>` | Specifies the routing instance that hosts the mediation devices. |
| `mutual-tls` | Global mutual TLS configuration. |
| `mutual-tls client authentication <certificate-name>` | (Optional) Name of the certificate. |
| `mutual-tls client authentication <certificate-name> certificate <certificate>` | (Optional) Specifies the certificate PEM data in base64 encoding. If this value is not specified, it defaults to the server certificate. |
| `mutual-tls client authentication <certificate-name> key-encrypted-text <key-encrypted-text>` | (Optional) Specifies the Certificate key in an encrypted format. |
| `mutual-tls client authentication <certificate-name> key-plain-text <key-plain-text>` | (Optional) Specifies the Certificate key in base64 encoding. If this value is not specified, it defaults to the server authentication key. |
| `mutual-tls client root-ca <root-ca>` | (Optional) Specifies the trusted CA in base64 encoding. This is mandatory for a self-signed certificate. |
| `mutual-tls server certificate <certificate>` | Specifies the certificate PEM data in base64 encoding. |
| `mutual-tls server client-ca <client-ca>` | Specifies the trusted client CAs in base64 encoding. |
| `mutual-tls server key <key-encrypted-text>` | Specifies the certificate key in base64 encoding. |
| `network-element-id <network-element-id>` | The network element ID of the network element in the ADMF. |
| `network-element-path <network-element-path>` | (Optional) The context-path for all incoming protocol requests. Default: /X1/NE. |
| `sync-timeout <5-60>` | (Optional) The maximum interval for completing protocol requests synchronously in seconds. Default: 5. |

| Attribute | Description |
|---|---|
| `async-timeout <10-120>` | (Optional) The maximum interval for completing protocol requests asynchronously in seconds. Default: 15 seconds. |
| `hold-time <60-86400>` | (Optional) Specifies the hold time in seconds for the LI tasks if no keepalive messages are seen from ADMF. Default: 3600 seconds.<br>If both sync and async-timeout are configured, async-timeout has to be at least twice the value of sync-timeout. |

**Invoking LI with CURL**

As shown in the example below, you can use CURL to invoke LI.

```
sudo curl --location 'http://localhost/hostconfd/api/v1/li' --header 'Content-Type:
application/json' --data '<Insert data>'  --unix-socket
/var/run/rtbrick/hostconfd_sock/unix.sock -v
```

**Enabling Lawful Interception**

RBFS restricts access to lawful interception data, making it accessible only to users with the highest privilege level or to explicitly defined LI operator roles.

RBFS supports lawful interception for PPPoE, L2TP, and IPoE subscribers.

### 5.7.3. RADIUS Lawful Interception

RBFS supports control of Lawful Interception (LI) through RADIUS, using either the Access-Accept or Change of Authorization (CoA) messages. To initiate or manage an LI session via RADIUS, all required LI attributes must be included in the RADIUS Access-Accept or CoA request.

These attributes are salt-encrypted using the algorithm defined in RFC 2868, originally specified for the Tunnel-Password attribute. As this encryption method is defined only for Access-Accept messages, RBFS uses a modified approach for CoA compatibility: the RADIUS Request Authenticator field is replaced with 16 zero bytes. This approach is widely adopted and compatible with multiple RADIUS server implementations.

RFC and draft compliance for RADIUS-based LI is partial unless explicitly stated otherwise.

VSA 26-50058-140 - RtBrick-LI-Action (salt encrypted integer)

| Value | Code | Description |
|---|---|---|
| NOOP | 0 | No action / Ignore LI request |
| ON | 1 | Start LI / Add LI request |

| Value | Code | Description |
|---|---|---|
| OFF | 2 | Stop LI / Delete LI request |

VSA 26-50058-141 - RtBrick-LI-Identifier (salt encrypted integer) Device unique lawful interception identifier (LIID) within the range from 1 to 4194303.

VSA 26-50058-142 - RtBrick-LI-Direction (salt encrypted integer)

| Value | Code | Description |
|---|---|---|
| INGRESS | 1 | Ingress mirroring only (from subscriber) |
| EGRESS | 2 | Egress mirroring only (to subscriber) |
| BOTH | 3 | Bidirectional mirroring (from and to the subscriber) |

VSA 26-50058-143 - RtBrick-LI-MED-Instance (salt encrypted string) The routing instance through which the mediation device is reachable.

VSA 26-50058-144 - RtBrick-LI-MED-IP (salt encrypted IPv4 address) IPv4 address of the mediation device.

VSA 26-50058-145 - RtBrick-LI-MED-Port (salt encrypted integer) UDP port between 49152 and 65535 set in the mirrored traffic

To prevent exposure of LI activity based on the presence of specific attributes, RBFS supports the special LI action NOOP. This action can be used to insert fake or decoy LI requests that have no operational effect but obfuscate detection attempts based on attribute presence alone.

For additional confidentiality, failed LI activations triggered via RADIUS do not return an error or NAK in the RADIUS response. This prevents external systems from distinguishing between valid and NOOP-based requests based on response behavior.

LI requests received via RADIUS, whether real or NOOP, are handled uniformly and appear in the same internal BDS table as requests provisioned through the X1 interface or the HTTP (REST) API.

### 5.7.4. RBFS Operational State API

The RBFS Operational State API provides endpoints for enabling and disabling LI on a per-subscriber basis:

- A HTTP POST request to `/subscribers/{subscriber_id}/enableLI?id={li_id}&direction={li_direction}&med_ip={med_ip}&med_instance={med_instance}&med_port={med_port}` enables LI for the specified subscriber
- A HTTP POST request to `/subscribers/{subscriber_id}/disableLI?id={li_id}` disable LI for the specified subscriber

The table below lists the request parameters:

*Request Parameters*

| Parameter Name | Description |
|---|---|
| `subscriber_id` | Subscriber identifier that is generated by RBFS, for example, 72339069014638701. |
| `id` | Identifier for Lawful Interception. This is a unique Identifier used by the mediation device to identify the intercepted subscriber. The range can be between 1 to 4194303. |
| `direction` | LI direction. Values are: INGRESS, EGRESS, BOTH. |
| `med_instance` | VRF instance through which the mediation device is reachable. |
| `med_ip` | IPv4 address of the mediation device |
| `med_port` | UDP port(MD)(49152-65535), mirrored traffic is forwarded |

All parameters are mandatory to enable LI.

**Request Examples**

**Enabling LI:**

The example below shows a curl command to enable LI:

> Specify the element name of your system instead of {element_name} in the preceding command.

```
curl -i -H "Content-Type: application/json" -X POST
'http://198.51.100.76:19091/api/v1/rbfs/elements/\{element_name}/services/opsd/proxy/s
ubscribers/72339069014639042/enableLI?id=66666&direction=BOTH&med_instance=libox&med_i
p=10.0.0.1&med_port=49153'
```

**Disabling LI:**

The example below shows a curl command to disable LI.

> Specify the element name of your system instead of {element_name} in the preceding command.

```
curl -i -H "Content-Type: application/json" -X POST
'http://198.51.100.76:19091/api/v1/rbfs/elements/\{element_name}/services/opsd/proxy/s
ubscribers/72339069014639042/disableLI?id=66666'
```

**Response Output**

The following is a sample output response for the `curl` command:

Output Response for enable & disable request:

```
HTTP/1.1 204 No Content
Date: Wed, 17 Sep 2025 06:23:47 GMT
Server: Python/3.10 aiohttp/3.8.1
```

In this example, the response code '204' is expected as LI request is successfully processed, but no return message will be received.

# 5.8. L2BSA

## 5.8.1. L2BSA Overview

Layer 2 Bitstream Access (L2BSA) refers to a scenario in which a service provider makes his access infrastructure available to other service providers. These are retail service providers who provide their Internet services. In Germany, this service is mandated by the Federal Network Agency (German: Bundesnetzagentur or BNetzA) which is the regulatory office for electricity, gas, telecommunications, post, and railway markets. It is a federal agency of the Federal Ministry for Economic Affairs and Energy and is headquartered in Bonn, Germany.

The definition of the L2BSA service was defined by the so-called NGA Forum, an advisory board founded in May 2010 by the Bundesnetzagentur for promoting dialogue between the Bundesnetzagentur, network operators, manufacturers, states and local authorities on NGA rollout.

https://www.bundesnetzagentur.de/DE/Fachthemen/Telekommunikation/Breitband/NGA-Forum/start.html

The L2BSA specification defines two interfaces, the so-called U interface (User Interface) at the customer location and the A10-NSP interface (A10 Network Service Provider) between the service provider networks. Those interface types were originally introduced in the Broadband Forum TR-101 (Migration to Ethernet-based Broadband Aggregation).

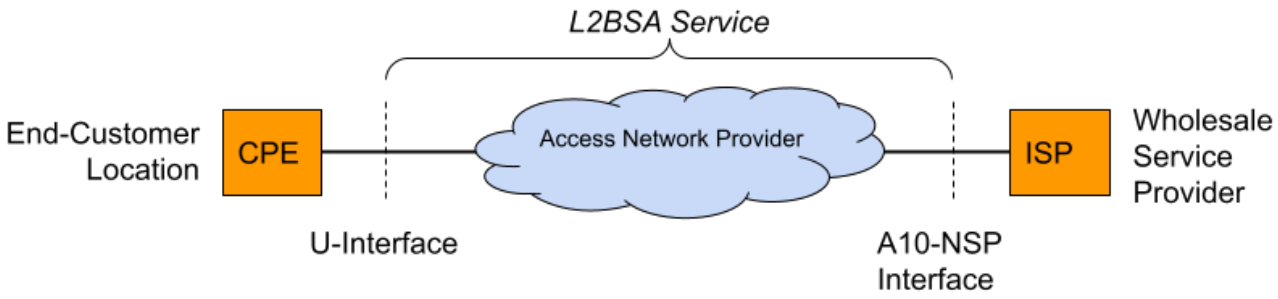



*Figure 92. L2BSA Interfaces*

The U interface is defined as a transparent Layer 2 interface which can be used with or without VLAN tags by the wholesale service providers. This means that some CPE will send their traffic untagged while another CPE is configured for tagged traffic which needs to be transparently forwarded between the U interface and the A10-NSP interface.

The A10-NSP interface is defined as a link aggregation bundle interface with one or more interfaces

and LACP enabled or disabled. All traffic on this interface is at least single-tagged with the so-called S-VLAN tag which identifies the U interface. This limits the amount of L2BSA services to 4094 per A10-NSP interface caused by the usable VLAN range. Therefore some providers need multiple A10-NSP interfaces if they need to address more than the 4094 services.

The term A10 relates to the end-to-end ADSL network reference model depicted in the figure below. The Core Network reference model is a subset of the end-to-end architecture; it is composed of two functional blocks and three reference points. The Access Node and the Regional Broadband Network are the two functional blocks. U, V and A10 are the three reference points.

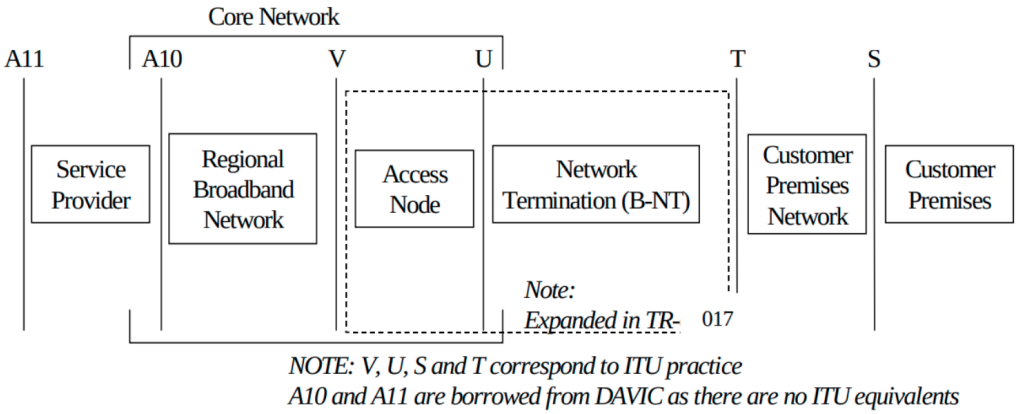



*Figure 93. TR-025*

Source: https://www.broadband-forum.org/download/TR-025.pdf

The mapping between the U interface and A10-NSP/S-VLAN is managed by the L2BSA service provider and typically changes frequently triggered by re-provisioning actions. Therefore all PPPoE discovery, as well as DHCPv4/v6 packets, must be enriched with additional line identification headers (Agent-Remote-Id, Agent-Circuit-Id, Actual-Data-Rate, ...) by the L2BSA service provider in the upstream direction (from U to A10-NSP interface) to allow the wholesale provider to identify the actual U interface for traffic received on the A10-NSP interface. This functionality is referred to as the intermediate agent functionality.

The U interface is terminated on the access node which might be an OLT for fiber-based access or MSAN for xDSL, therefore the U interface can also be called the customer access line or port. The access node will add additional VLAN for each access line to uniquely identify them between the access node and leaf switch. This VLAN is called the ANP tag (Access-Node-Port).

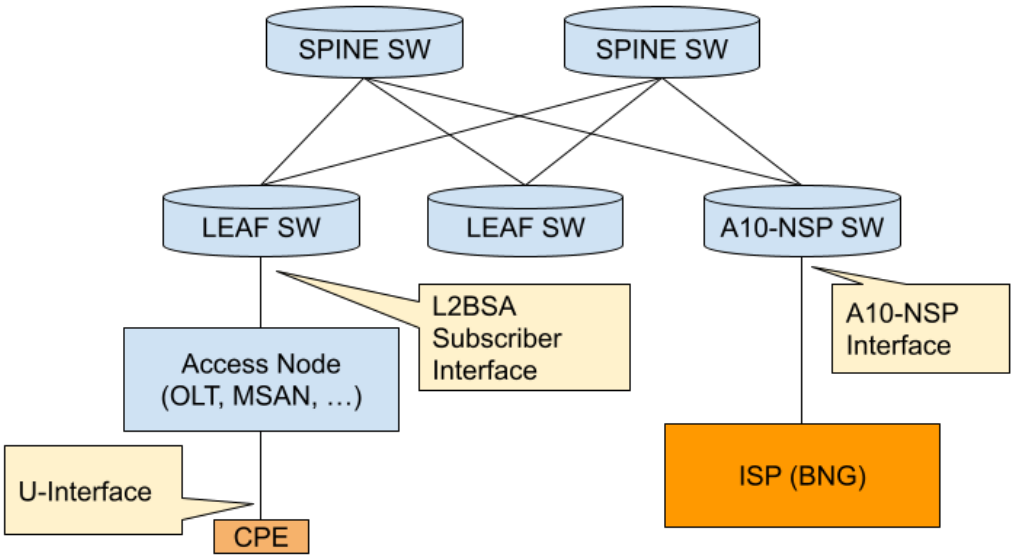



*Figure 94. L2BSA Fabric*

For L2BSA services all traffic received with a given ANP tag will be transparently forwarded to the corresponding A10-NSP interface via Layer 2 cross-connect (L2X) services. We introduce a new subscriber type for L2BSA in our RBFS access infrastructure to manage those services on the leaf switches and to allow further advanced access services like QoS or subscriber accounting.

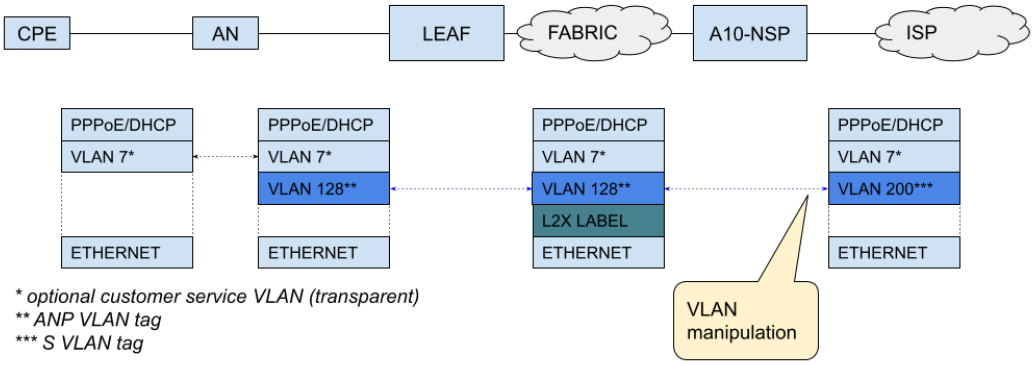



*Figure 95. L2BSA Traffic*

### L2BSA Services and Subscribers

RBFS distinguishes between L2BSA services and subscribers. L2BSA services are created using the Operational State API (`/l2bsa`) stored as ephemeral objects in the configuration daemon. Those objects are excluded from the actual configuration and not restored after reboot. The L2BSA service represents a defined interface to add, update and delete L2BSA services. Those services are managed by external orchestrators or service managers and are not changed or deleted by RBFS.

Adding such an L2BSA service triggers the creation of a corresponding L2BSA subscriber which represents the internal state. Each L2BSA subscriber is a full-access subscriber and therefore most subscriber features will work for L2BSA as well. This includes features like RADIUS authentication, accounting, CoA or dynamic QoS. Each L2BSA subscriber persists as long as the service is still present and also if the subscriber has already been terminated. This allows external applications to collect all states and counters before the subscriber is finally deleted triggered by L2BSA service delete.

#### A10-NSP L2X Services

For each L2BSA service, there must also be an A10-NSP L2X endpoint created using the Operational State API (`/a10nsp/l2x`).

The A10-NSP endpoint is the place where all VLAN manipulation is done meaning where ANP and S-VLAN are swapped.

##### VLAN Range Support for L2BSA Subscribers

You can use S-VLAN range for L2BSA subscribers on A10NSP (QAX) platform. It allows you to define a range of VLANs instead of configuring each VLAN separately. The related API endpoints described in the API guide.

#### Supported Platforms

Support for the following underlays are available:

- IS-IS Segment Routing
- BGP Segment Routing

> Not all features are necessarily supported on each hardware platform. Refer to the Platform Guide for the features and the sub-features that are or are not supported by each platform.

### 5.8.2. L2BSA Configuration

This document assumes a working base configuration explaining the L2BSA specific additions only.

#### AAA Profile Configuration

Each L2BSA service refers to a mandatory AAA configuration profile as explained in detail in the Subscriber Management Configuration Guide. The following example shows the minimum configuration with authentication and accounting disabled.

```
supervisor@leaf1: cfg> show config access aaa-profile aaa-l2bsa-default
{
    "rtbrick-config:aaa-profile": [
      {
        "profile-name": "aaa-l2bsa-default",
        "authentication": {
          "order": "NONE"
        },
        "accounting": {
          "order": "NONE"
        }
      }
    ]
  }
```

### 5.8.3. L2BSA Operational Commands

The following commands allow to list or clear all L2BSA services from CLI.

L2BSA subscribers can be managed like any other subscriber using the subscriber CLI commands.

Example: L2BSA service

```
supervisor@rtbrick: op> show l2bsa service
Interface         ANP VLAN      Timestamp
lag-1             101           Fri Jun 05 12:28:46 GMT +0000 2026
lag-1             102           Fri Jun 05 12:28:46 GMT +0000 2026
lag-1             103           Fri Jun 05 12:28:46 GMT +0000 2026
lag-1             104           Fri Jun 05 12:28:46 GMT +0000 2026
lag-1             105           Fri Jun 05 12:28:46 GMT +0000 2026
lag-1             106           Fri Jun 05 12:28:46 GMT +0000 2026
lag-1             107           Fri Jun 05 12:28:46 GMT +0000 2026
lag-1             108           Fri Jun 05 12:28:46 GMT +0000 2026
lag-1             109           Fri Jun 05 12:28:46 GMT +0000 2026
lag-1             110           Fri Jun 05 12:28:46 GMT +0000 2026
```

Example: L2BSA subscriber

```
supervisor@rtbrick: op> show subscriber
Subscriber-Id           Interface         VLAN       Type    State
281479271677953         lag-1             101:0      L2BSA   ESTABLISHED
281479271677954         lag-1             102:0      L2BSA   ESTABLISHED
281479271677955         lag-1             103:0      L2BSA   ESTABLISHED
281479271677956         lag-1             104:0      L2BSA   ESTABLISHED
281479271677957         lag-1             105:0      L2BSA   ESTABLISHED
281479271677958         lag-1             106:0      L2BSA   ESTABLISHED
281479271677959         lag-1             107:0      L2BSA   ESTABLISHED
281479271677960         lag-1             108:0      L2BSA   ESTABLISHED
281479271677961         lag-1             109:0      L2BSA   ESTABLISHED
281479271677962         lag-1             110:0      L2BSA   ESTABLISHED
```

Example: L2BSA Subscriber filter type

```
supervisor@rtbrick: op> show subscriber filter type L2BSA
Subscriber-Id           Interface         VLAN       Type    State
281479271677953         lag-1             101:0      L2BSA   ESTABLISHED
281479271677954         lag-1             102:0      L2BSA   ESTABLISHED
281479271677955         lag-1             103:0      L2BSA   ESTABLISHED
281479271677956         lag-1             104:0      L2BSA   ESTABLISHED
281479271677957         lag-1             105:0      L2BSA   ESTABLISHED
281479271677958         lag-1             106:0      L2BSA   ESTABLISHED
281479271677959         lag-1             107:0      L2BSA   ESTABLISHED
281479271677960         lag-1             108:0      L2BSA   ESTABLISHED
```

```
281479271677961         lag-1             109:0      L2BSA   ESTABLISHED
281479271677962         lag-1             110:0      L2BSA   ESTABLISHED
```

Example: L2BSA Subscriber details for a specified subscriber

```
supervisor@rtbrick: op> show subscriber 281479271677953 detail
Subscriber-Id: 281479271677953
    Type: L2BSA
    State: ESTABLISHED
    Created: Fri Jun 05 12:28:47 GMT +0000 2026
    Interface: lag-1
    Outer VLAN: 101 (802.1q)
    IFL: l2bsa-lag-1/281479271677953
    Username: lag-1:101@l2bsa
    Agent-Remote-Id: DEU.DTAG.L2BSA201
    Agent-Circuit-Id: 0.0.0.0/0.0.0.0 eth 201
    AAA-Profile: L2BSA-AAA
    Session-Timeout: 0 seconds (disabled)
    Idle-Timeout: 0 seconds (disabled)
    L2X:
        Instance: default
        Nexthop: 192:1::1 (ipv6/labeled-unicast)
        Ingress Label (TX): 10001
        Egress Label (RX): 10001
    Accounting:
        Session-Id: 281479271677953:1780662527
        Start-Time: 2026-06-05T12:28:47.237107+0000
        Interims Interval: 0 seconds (disabled)
```

Example for clear L2BSA service command.

```
supervisor@leaf1: op> clear l2bsa service all
Success
```

### 5.8.4. L2BSA Operational State API

The API is split into the L2BSA and A10-NSP L2X service API. Details about those API endpoints, schema and attributes can be found in the official RBFS Operational State OpenAPI documentation.

#### L2BSA Services API

Each L2BSA service is uniquely identified by physical interface name (IFP) and ANP VLAN identifier. All other attributes of L2BSA services can be changed dynamically by replacing the existing service object by doing a full update meaning that updates must include all attributes and RBFS will automatically recognize the changes to the last version to apply those incrementally.

| Endpoint | Description |
|---|---|
| GET /l2bsa | This endpoint returns a list of all L2BSA services. |
| PUT /l2bsa | This endpoint replaces all L2BSA services on the system by adding new ones, updating existing and deleting those not included in the request but present on the system. Therefore this endpoint can be used to delete all services if an an empty list is provided in the request body. |
| GET /l2bsa/{ifp_name} | This endpoint returns a list of all L2BSA services on the given physical interface (IFP). |
| PUT /l2bsa/{ifp_name} | This endpoint works similarly to `PUT /l2bsa` but affects only services matching the given physical interface (IFP). Therefore this endpoint can be used to delete all services of a given physical interface if an empty list is provided in the request body. |
| GET /l2bsa/{ifp_name}/{anp} | This endpoint returns a single L2BSA service or `404` if no matching service is found. |
| PUT /l2bsa/{ifp_name}/{anp} | This endpoint adds or updates a single L2BSA service. |
| DELETE /l2bsa/{ifp_name}/{anp} | This endpoint deletes a single L2BSA service. |

The following example shows an L2BSA service request body with only the mandatory attributes set.

```
{
    "ifp_name": "ifp-0/0/1",
    "anp_vlan": 128,
    "aaa_profile_name": "aaa-l2bsa-default",
    "l2x": {
      "ingress_nexthop": "2001:db8:0:30::",
      "ingress_service_label": 1001,
      "egress_service_label": 1002
    }
}
```

The L2X ingress next-hop is typically the fabric loopback IPv4 or IPv6 address of the corresponding switch where the corresponding A10-NSP L2X service is placed. The ingress label is the upstream service label added for traffic sent to the corresponding next-hop (sent label). The egress label is the downstream service label where traffic sent from the next-hop is expected to be received (receive label). This label must be unique per switch.

Only labels greater than 100.000 should be used for the L2X service labels because labels between 16 - 100.000 are reserved by other applications. L2X service labels must not conflict with other L2X service labels and the same label can be used for downstream and upstream.

L2BSA supports also dynamic QoS which can be controlled by RADIUS if enabled or directly via API as shown in the following example.

```
{
    "ifp_name": "ifp-0/0/1",
    "anp_vlan": 128,
    "aaa_profile_name": "aaa-l2bsa-default",
    "l2x": {
      "ingress_nexthop": "2001:db8:0:30::",
      "ingress_service_label": 1001,
      "egress_service_label": 1002
    },
    "qos": {
      "qos_profile_name": "l2bsa-qos-default",
      "parent_scheduler": "pon1",
      "shaper": "name=shaper_session,high=14000,low=2000;name=shaper_voice,high=2000",
      "policer": "level=4,cir=1m,cbs=256;level=3,cir=2m,cbs=512",
      "queue": "name=BestEffort,size=65000;name=Voice,size=20000"
    }
}
```

The QoS configuration profile is used to instantiate the QoS resources for the subscriber including schedulers, queues and shapers. This profile can be set using the `qos_profile_name` attribute. All dynamic QoS settings like MFC, queue sizes, shaper and policer rates will be reset if this attribute is present also if not changed. Additional shaper or policer settings included will be applied to the new QoS configuration after reset.

The assigned shaper instances can be updated using the `shaper` attribute which will apply to the QoS instance of the corresponding subscriber-only, but not to the other subscribers using the same QoS profile. It is only possible to update existing shapers dynamically but it is not possible to create a new shaper.

```
"shaper": "<shaper-name>,<high-kbps>,<low-kbps>;<shaper-name>,…"
```

This attribute can be also used as a key-value list which is automatically recognized by RBFS.

```
"shaper": "name=<shaper-name>,high=<high-kbps>,low=<low-kbps>;…"
```

The `policer` attribute is also a string that contains a list of multiple policer-level settings separated by a semicolon. Each set contains a level, cir, cbs, pir, pbs, max-cir and max-pir separated by a comma.

```
"policer": "<level>,<cir>,<cbs>,<pir>,<pbs>,<max-cir>,<max-pir>;<level>…"

Example:
"policer": "1,2000,200;2,8000,800;3,0,800;4,0,800"
```

```
level: 1 (highest priority) to 4 (lowest priority)
cir: ingress policer committed information rate (kbps)
cbs: ingress policer committed burst size (kbps)
pir: ingress policer peak information rate (kbps)
pbs: ingress policer peak burst size (kbps)
max-cir: max ingress policer committed information rate (kbps)
max-pir: max ingress policer peak information rate (kbps)
```

If PIR and PBS are not defined, the values from CIR and CBS are used as PIR and PBS as well. The max CIR and max PIR attributes are optional default set to unlimited.

This attribute can be also used as a key-value list which is automatically recognized by RBFS.

```
"policer": "level=<level>,cir=<cir>,cbs=<cbs>,pir=<pir>,pbs=<pbs>,max-cir=<max-
cir>,max-pir=<max-pir>;…"

Example:
"policer": "level=4,cir=1m,cbs=256;cir=2m,cbs=512,level=3"
```

The `queue` attribute is a string that contains a list of multiple queue settings separated by a semicolon. Each queue setting contains a queue name and queue size in bytes separated by a comma.

```
"queue": "<queue-name>,<size-bytes>;<queue-name>,<size-bytes>;…"
```

This attribute can be also used as a key-value list which is automatically recognized by RBFS.

```
"queue": "name=<queue-name>,size=<size-bytes>;name=…"
```

> The subscriber management infrastructure does not check if a referenced QoS profile, shaper, queue or policer exists or not. This is handled by forwarding infrastructure which continues processing the subscriber QoS settings as soon as the missing resource was added.

The parent scheduler element of the scheduler map assigned to the subscriber can be selected with the `parent_scheduler` attribute. If not present, the scheduler map will be directly bound to the local IFP where the session is established. The parent scheduler can be only set at the beginning and must not be changed.

> Providing a QoS parent scheduler that is not present on the corresponding IFP will lead to the black howling of all egress data traffic.

The optional `access_line_info` attribute contains the access line identification. A detailed list of all supported line attributes and values can be found in the official RBFS Operational State OpenAPI

documentation.

> For access line attributes there is a difference between attributes not present or set to zero.

**Example:**

```
/usr/bin/curl --location --request PUT
'http://198.51.100.35:19091/api/v1/rbfs/elements/leaf1/services/opsd/proxy/l2bsa/ifp-
0%2f0%2f1/128' --header 'Content-Type: application/json' --data-raw '{
    "ifp_name": "ifp-0/0/1",
    "anp_vlan": 128,
    "aaa_profile_name": "aaa-l2bsa-default",
    "l2x": {
        "ingress_nexthop": "2001:db8:0:30::",
        "ingress_service_label": 1001,
        "egress_service_label": 1002
    },
    "qos": {
        "qos_profile_name": "l2bsa-qos-default",
        "parent_scheduler": "pon1",
        "shaper":
"name=shaper_session,high=14000,low=2000;name=shaper_voice,high=2000",
        "policer": "level=4,cir=1m,cbs=128;level=2,cir=1m,cbs=128",
        "queue": "name=BestEffort,size=65000;name=Voice,size=20000"
    },
    "access_line_info": {
        "agent_circuit_id": "0.0.0.0/0.0.0.0 eth 0/0",
        "agent_remote_id": "DEU.RTBRICK.L2BSA01",
        "upstream": {
            "actual_rate": 2048
        },
        "downstream": {
            "actual_rate": 16235
        },
        "pon_type": "GPON"
    }
}'
```

**L2BSA Subscribers API**

The existing subscriber API was extended to do some common actions using the L2BSA physical interface name (IFP) and ANP VLAN identifier instead of the subscriber-id.

| Endpoint | Description |
| --- | --- |
| GET /subscribers/l2bsa/{ifp_name}/{anp} | This endpoint is an alias for `GET /subscribers/{subscriber_id}` returning detailed subscriber information or `404` if no matching subscriber is found. |
| DELETE /subscribers/l2bsa/{ifp_name}/{anp} | This endpoint is an alias for `DELETE /subscribers/{subscriber_id}` which terminates a matching L2BSA subscriber. |
| GET /subscribers/l2bsa/{ifp_name}/{anp}/adjusted-accounting | This endpoint is an alias for `GET /subscribers/{subscriber_id}/adjusted-accounting` returning the adjusted accounting information or `404` if no matching subscriber is found. |

**A10-NSP L2X Services API**

| Endpoint | Description |
| --- | --- |
| GET /a10nsp/l2x | This endpoint returns a list of all A10-NSP L2X services. |
| PUT /a10nsp/l2x | This endpoint replaces all A10-NSP L2X services on the system by adding new ones, updating existing and deleting those not included in the request but present on the system. Therefore this endpoint can be used to delete all services if an an empty list is provided in the request body. |
| GET /a10nsp/l2x/{lag_interface_name} | This endpoint returns a list of all A10-NSP L2X services on the given the LAG interface. |
| PUT /a10nsp/l2x/{lag_interface_name} | This endpoint works similarly to `PUT /a10nsp/l2x` but affects only services matching the given LAG interface. Therefore this endpoint can be used to delete all services of a given the LAG interface if an empty list is provided in the request body. |
| GET /a10nsp/l2x/{lag_interface_name}/{s_anp} | This endpoint returns a single A10-NSP L2X service or `404` if no matching service is found. |
| PUT /a10nsp/l2x/{lag_interface_name}/{s_anp} | This endpoint adds or updates a single A10-NSP L2X service. |
| DELETE /a10nsp/l2x/{lag_interface_name}/{s_anp} | This endpoint deletes a single A10-NSP L2X service. |

The following example shows an A10-NSP L2X service request body with only the mandatory attributes set.

```
{
    "lag_interface_name": "lag-1",
```

```
    "s_vlan": 100,
    "anp_vlan": 128,
    "l2x": {
        "ingress_nexthop": "2001:db8:0:40::",
        "ingress_service_label": 1002,
        "egress_service_label": 1001
    }
}
```

The L2X ingress next-hop is typically the fabric loopback IPv4 or IPv6 address of the corresponding switch where the corresponding L2BSA service is placed. The ingress label is the upstream service label added for traffic sent to the corresponding next-hop (send label). The egress label is the downstream service label where traffic sent from the next-hop is expected to be received (receive label). This label must be unique per switch.

> Only labels greater than 100000 should be used for the L2BSA service labels because labels between 16 - 100000 are reserved by other services. L2BSA service labels must not conflict with other L2X service labels and the same label can be used for downstream and upstream.

**Example:**

```
/usr/bin/curl --location --request PUT
'http://198.51.100.35:19091/api/v1/rbfs/elements/a10nsp-
sw1/services/opsd/proxy/a10nsp/l2x/lag-1/100' \
--header 'Content-Type: application/json' \
--data-raw '{
    "lag_interface_name": "lag-1",
    "s_vlan": 100,
    "anp_vlan": 128,
    "l2x": {
        "ingress_nexthop": "2001:db8:0:40::",
        "ingress_service_label": 1002,
        "egress_service_label": 1001
    }
}'
```

> - L2BSA supports both PPPoE and IPoE subscribers.
> - L2BSA supports DHCPv4 and DHCPv6 protocols.

## 5.9. Redundancy

Redundancy is a high-availability solution that ensures uninterrupted network service by enabling seamless control plane failover and rapid recovery. RBFS supports Redundancy for both IPoE and PPPoE subscribers. The Redundancy framework provides redundancy mechanisms to maintain subscriber services during node failures or switchover events.

RBFS supports the following Redundancy models for IPoE and PPPoE subscribers.

For IPoE subscribers, RBFS supports:

- RD-Based Stateful Redundancy (Supports both Active-Active and Active-Standby). For details, see RD-Based Stateful Redundancy for IPoE Subscribers.
- Stateless Redundancy (Supports Active-Standby model). For details, see Stateless Redundancy for IPoE Subscribers.

For PPPoE Subscribers, RBFS supports:

- RD-Based Stateless Redundancy. For details, see RD-Based Stateless Redundancy for PPPoE Subscribers.
- PADO Delay-based Redundancy (Stateless). For details, see PADO Delay-based Redundancy

The following diagram illustrates the RBFS Redundancy models for various deployments.

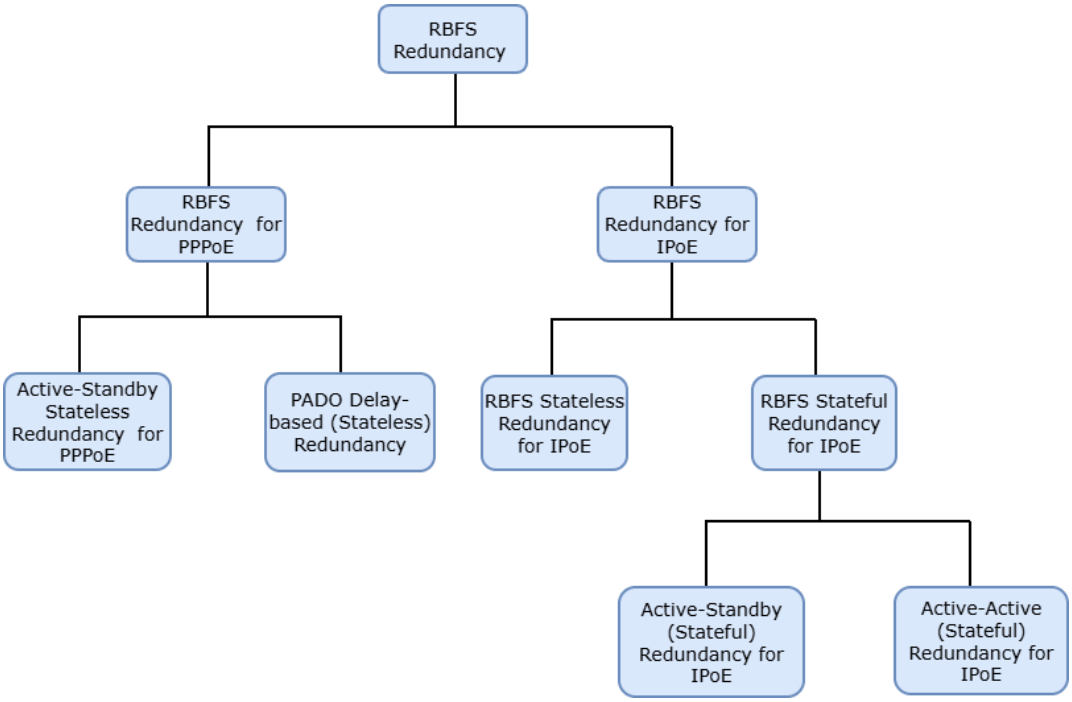



*Figure 96. RBFS Redundancy Models*

## 5.9.1. Stateful vs Stateless Redundancy for IPoE

**Stateful Redundancy**:

Stateful redundancy maintains full subscriber session state synchronized between the redundant nodes. In the event of a failure, the standby node takes over seamlessly without requiring subscriber re-authentication or session re-establishment.

Key Benefits:

- Almost seamless failover with minimal traffic disruption.
- Subscribers do not need to renew DHCP or restart sessions.
- Maintains accounting, QoS, and policy information.

When to Use:

- For high-availability environments where packet loss must be minimal.
- When re-authentication or changes in IP address are not acceptable.

**Stateless Redundancy**:

Stateless Redundancy does not synchronize subscriber session state between the nodes. When an outage occurs on the active node, all existing IPoE sessions on that node are terminated. Affected subscribers are required to re-initiate abd re-establish the sessions.

Key Benefits:

- Simple design
- Easy to deploy and manage.

When to Use:

- Deployments where brief service interruptions are acceptable.
- Networks that prefer scalability and simplicity over seamless failover.
- Cost-sensitive networks with limited resources.

### 5.9.2. Licensing

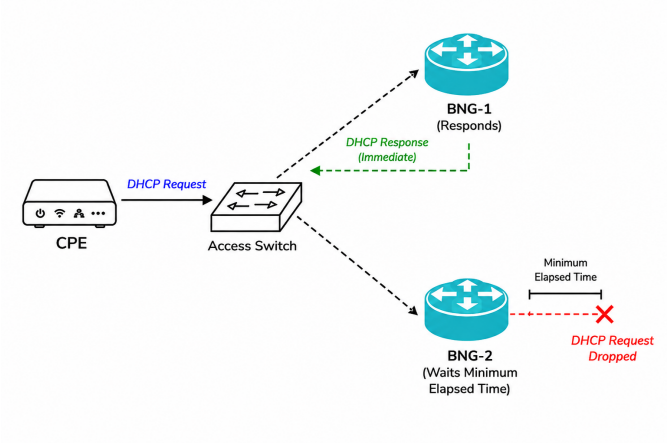
- RD-Based Redundancy is supported only with the **Benefits** licensing for subscribers.
- All other redundancy modes are supported with both **Essentials** licensing for subscribers and **Benefits** licensing for subscribers.

### 5.9.3. Stateless Redundancy for IPoE Subscribers

Stateless Redundancy uses active-standby node model. Only one BNG node is active at a time, handling all subscriber sessions. The standby BNG node stays ready to take over if the active node fails.

Both the active and standby BNG nodes receive the same DHCP broadcast messages from clients for IPv4. When a client sends a DHCPDISCOVER message, both BNG nodes receive it. The active BNG processes and responds right away. The standby BNG node, configured with a non-zero DHCP minimum elapsed time ignores requests if the elapsed time is below the set threshold.

If the active BNG does not respond in time, the client re-sends the DHCP request with an updated elapsed time that meets the threshold. The standby BNG node then processes the request and performs the subscriber service, ensuring continuous service.




### DHCP Minimum Elapsed Time

The system leverages the DHCP minimum elapsed time technique. For DHCPv4, the 'secs' field in DHCPDISCOVER and DHCPREQUEST messages is compared to the configured threshold. For DHCPv6, the elapsed time option in SOLICIT messages is used. Messages below the threshold are dropped.

This method ensures only one BNG node responds at a time and allows automatic failover if the active becomes unavailable.

By default, the minimum elapsed time is '0' seconds. This feature is supported in all DHCP operating modes which include 'server', 'relay', and 'proxy'.

> The DHCP minimum elapsed-time configuration option is available at both the Access Interface and DHCP Relay configurations for DHCPv4 and DHCPv6.

The following example configures double-tagged interface 'ifp-0/0/0' for IPoE access and sets DHCP minimum elapsed time to 5 seconds.

```
set access interface double-tagged ifp-0/0/0 access-type IPoE
set access interface double-tagged ifp-0/0/0 dhcp-min-elapsed-time 5
```

```
{
  "rtbrick-config:double-tagged": {
    "interface-name": "ifp-0/0/0",
    "access-type": "IPoE",
    "dhcp-min-elapsed-time": 5
  }
}
```

> In IPoE Stateless Redundancy deployments, `rapid-commit` must be set to 'false' to ensure the standard four-message DHCPv6 exchange is used. This is required to prevent dual-stack splitting and ensure correct subscriber session establishment.

For Rapid Commit configuration information, see DHCPv6 Rapid Commit Control.

**Dual-Stack Splitting Prevention in IPoE Stateless Redundancy Deployments**

This feature prevents dual-stack splitting by delaying DHCPv6 session establishment until a valid DHCPv4 binding exists for the subscriber in IPoE Stateless Redundancy deployments. This ensures that both IPv4 and IPv6 sessions are established on the same BNG, maintaining a consistent dual-stack subscriber session in stateless redundancy deployments.

In Stateless Redundancy for IPoE deployments, DHCPv4 and DHCPv6 requests may be processed by different BNGs, resulting in dual-stack splitting. For example, consider a dual-stack IPoE subscriber who requires both DHCPv4 and DHCPv6 connectivity. In a Stateless Redundancy deployment, the DHCPv4 request may be processed by BNG-A, while the DHCPv6 request is processed by BNG-B. As a result, the subscriber's IPv4 and IPv6 sessions are established on different gateways, creating a dual-stack split.

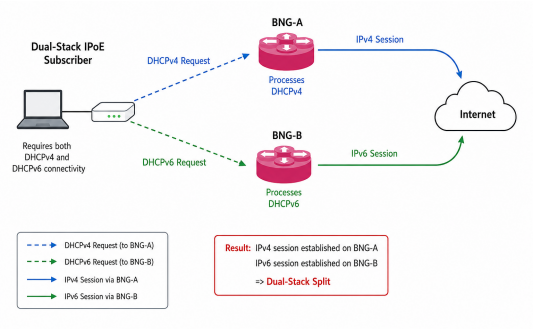



*Figure 97. Dual-stack split example for IPoE Stateless Redundancy*

When `wait-for-dhcpv4` is enabled, DHCPv6 session establishment is deferred until a valid DHCPv4 binding is present. It ensures that both IPv4 and IPv6 sessions are anchored to the same gateway.

For information about configuration commands, see Configuring DHCPv6.

Example:

The following configuration enables DHCPv4 session establishment before processing DHCPv6 requests by setting `wait-for-dhcpv4` to `true`. It sets the DHCPv6 lease lifetime to 90 seconds.

```
set access access-profile ipoe protocol dhcpv6 wait-for-dhcpv4 true
```

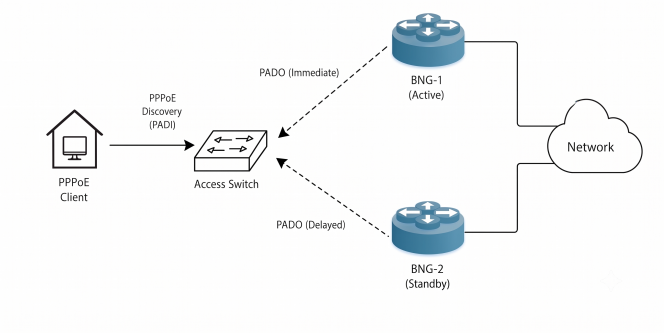
```
{
  "rtbrick-config:dhcpv6": {
    "enable": "true",
    "wait-for-dhcpv4": "true",
    "lifetime": 90
  }
}
```

## 5.9.4. PADO Delay-Based Redundancy for PPPoE Subscribers

RBFS supports a PADO delay-based stateless redundancy method for PPPoE subscribers. By adding a configurable delay to PADO responses, multiple BNGs can participate in discovery while allowing

a preferred BNG to respond first. This approach provides redundancy without requiring subscriber state synchronization between the BNG nodes. In the event of active BNG failure, subscriber sessions are re-established on the standby BNG as new sessions, with re-convergence typically taking about 90 seconds.

The following image illustrates the PPPoE client sending a PADI discovery message that is received by both BNGs through the access switch. The active BNG promptly responds with a PADO message, while the standby BNG, configured with the minimum elapsed time, delays its PADO response.




The `pppoe-pado-delay` command option must be configured only on the standby BNG. The delay prevents the standby node from responding before the active BNG during PPPoE subscriber discovery.

Example:

This configuration enables PPPoE access on the double-tagged VLAN range on interface 'ifp-0/0/0', using the specified access and AAA profiles for subscriber authentication and session management. It also configures a PPPoE PADO delay of 30 milliseconds, allowing the Standby BNG to wait before responding to subscriber discovery requests.

```
set access interface double-tagged ifp-0/0/0 201 250 201 250
set access interface double-tagged ifp-0/0/0 201 250 201 250 access-type PPPoE
set access interface double-tagged ifp-0/0/0 201 250 201 250 access-profile-name
pppoe-default-ds
set access interface double-tagged ifp-0/0/0 201 250 201 250 aaa-profile-name aaa-
profile
set access interface double-tagged ifp-0/0/0 201 250 201 250 pppoe-pado-delay 30
```

```
{
  "rtbrick-config:double-tagged": {
    "interface-name": "ifp-0/0/0",
    "outer-vlan-min": 201,
    "outer-vlan-max": 250,
    "inner-vlan-min": 201,
    "inner-vlan-max": 250,
    "access-type": "PPPoE",
    "access-profile-name": "pppoe-default-ds",
    "aaa-profile-name": "aaa-profile",
    "pppoe-pado-delay": 30
```

```
  }
}
```

### 5.9.5. Stateful Redundancy for IPoE Subscribers

**Overview**

Node outages and link failures that may occur on an access network can bring down the subscriber services. These network outages affect critical workloads and continuity of business. So, it is essential to set up a network that is resilient and responds quickly to the events and protects the network from outages.

The following diagram represents a simple access network without redundancy.

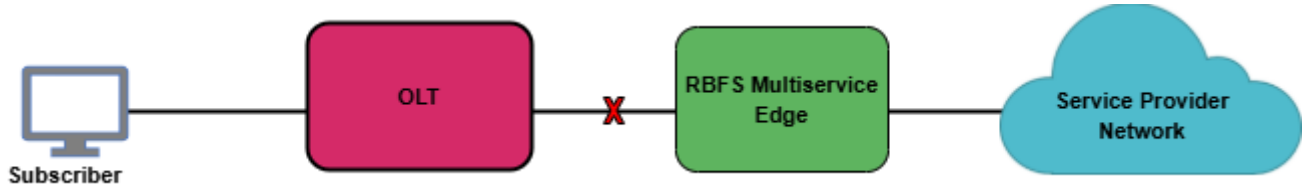



*Figure 98. Access Network Without Redundancy*

IPoE RD-based Stateful Redundancy protects subscriber services from node or link outages. RD-based redundancy refers to a deployment that uses the Redundancy Daemon (RD), an application designed to manage redundancy deployment on the access side of the BNG, especially in topologies similar to Multi-Chassis LAG.

It provides mechanisms to enhance network resiliency that enables subscriber workloads to remain functional by ensuring a reliable switchover in the event of a node or link outage. With Redundancy, if one node goes down due to node or link failure, another node can automatically take over the services.

- RBFS currently supports stateful redundancy only for IPoE subscribers.
- The term 'Multiservice Edge' used in this document refers to a Multiservice Edge platform. The Multiservice Edge platform integrates all BNG functionalities within a single platform. In contrast, the spine-leaf topology distributes these functionalities between spine and leaf platforms based on their respective roles.
- IPoE RD-based Stateful Redundancy is supported only on Multiservice Edge platforms. This feature is not supported on spine-leaf network topology as the spine-leaf topology architecture itself innately provides a redundant and resilient network.
- For IPoE RD-based Stateful Redundancy, all subscribers must reside in the same routing instance. Assigning subscribers to different routing instances, either statically or dynamically via RADIUS, is currently not supported.

### Understanding RD-Based Stateful Redundancy for IPoE

RBFS supports stateful redundancy and fault tolerance through two primary mechanisms:

- Active-Active Redundancy
- Active-Standby Redundancy

You can select the desired redundancy mechanism and configure the system accordingly to enable the feature. By default, the Active-Active (`Standby-Forwarding`) mode is enabled. If you prefer to use the Active-Standby Redundancy mode, select the `Standby-Non-Forwarding` option in the Platform Profile configuration. This is a one time configuration unless you choose to switch to the other Redundancy mode.

IPoE RD-based Stateful Redundancy is centered around subscriber groups, known as redundancy sessions. It protects subscriber groups using a two-node cluster model, either active-active or active-standby.

The following image illustrates two different redundancy mechanisms: Active-Active and Active-Standby. In Active-Active mode, both BNG nodes and LAGs forward traffic. In Active-Standby mode, only the active node and the LAG (between the OLT and the active node) handle traffic, while the standby node remains idle until a failover occurs.

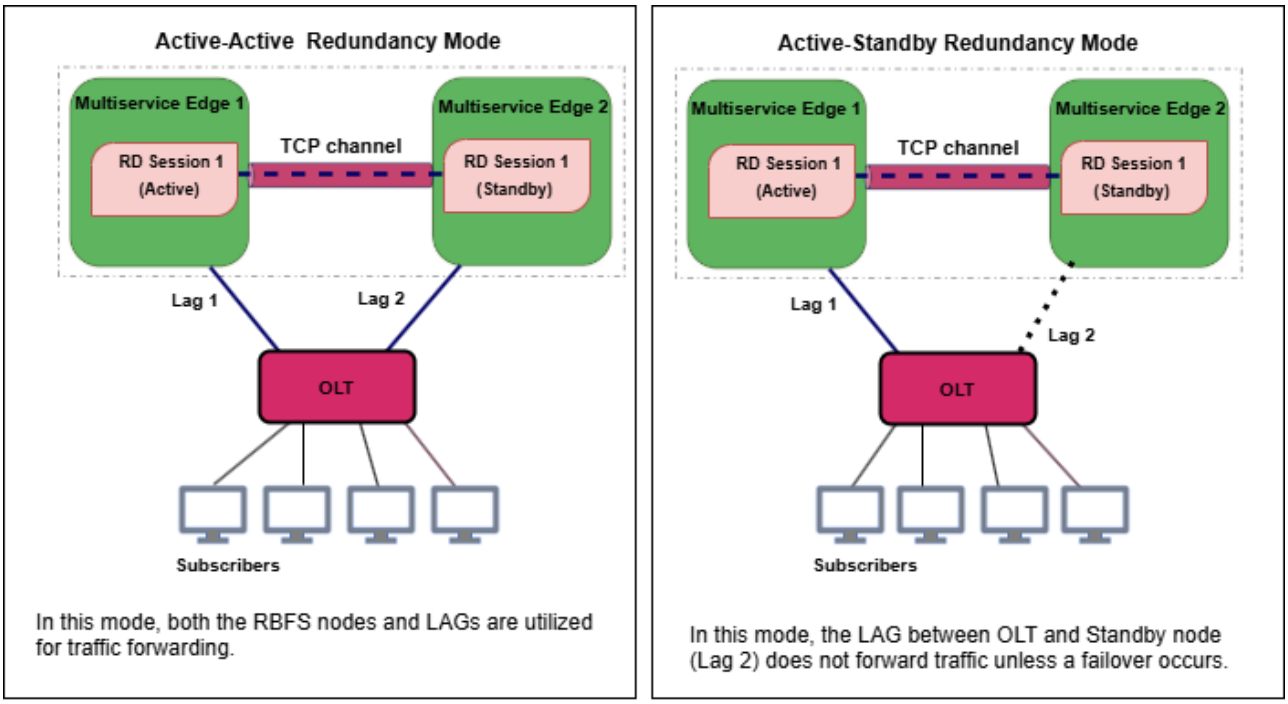



*Figure 99. Active-Active and Active-Standby Redundancy Mechanisms*

From the perspective of the OLT, there is only one LAG that is distributed across multiple BNGs. This means that the OLT sees one logical link, even though it is physically connected to two separate BNGs. But, in reality, this LAG spans two BNGs, indicating that the OLT is actually connected to two different devices.

#### Redundancy Session

Redundancy session is a replication mechanism that is used to pair Multiservice Edge nodes for redundancy. Redundancy allows grouping of subscribers, under a redundancy session and each

redundancy session is represented by a redundancy session ID. Simply put, an redundancy session represents a redundancy group of subscribers. A redundancy session enables linking the LAG with that particular redundancy session (subscriber group). When you define a value for the redundancy session ID, this ID should be unique and the same for both redundancy pairs. When two nodes get the same session ID, they recognize each other as the peer nodes for a particular redundancy session (subscriber group). The TCP session establishment between the nodes occurs after the pairing with the redundancy session ID. Once the TCP session is established, the nodes use this channel for subscriber data mirroring and synchronization and for health status monitoring.

A Multiservice Edge chassis can contain multiple redundancy sessions. Multiple Multiservice Edge nodes can be active nodes for one or more subscriber groups (redundancy sessions) that serve the subscribers and they can, at the same time, be standby nodes for other subscriber groups.

One node, which is paired for redundancy, can perform subscriber services for more than one redundancy session (subscriber group). The peer node, which is identical to the first node, contains the same subscriber group (redundancy session) as a standby. And in the event that first node goes down due to any outage, the standby node can take over subscriber service for this redundancy session.

RD-based IPoE Stateful Redundancy allows running a redundancy session actively on one node and backing up the same redundancy session (subscriber group) on a different (standby) Multiservice Edge node. In IPoE redundancy, in fact, there is no active node or standby node. It is either an active subscriber group or standby subscriber group. A Multiservice Edge node can be active for a subscriber group and at the same time it can be a standby for a different subscriber group. You can configure a maximum number of 64 redundancy sessions (either active or standby) on a Multiservice Edge node.

#### Active-Active Redundancy

In the Active-Active Redundancy mode, both of the nodes and their associated LAGs are actively used for traffic forwarding. Both the nodes synchronize their subscriber state information in real-time. This ensures that all the LAG links towards the OLT are utilized for maximum resource efficiency.

If one node fails, the other node can immediately take over the responsibilities with minimal disruption, as it is already synchronized with the active node. This mechanism provides fast failover with minimal traffic disruption and ensures traffic flow.

#### Pinning or Assigning a Subscriber to a Link of a LAG

RBFS uses a pinning mechanism where the system assigns or 'pins' each subscriber to each of the LAG member links between active/standby nodes and the OLT.

Subscribers are assigned during session bringup to member links belonging to the access LAG on both of the nodes. They will be assigned either as locally pinned or remotely pinned. This ensures an equal distribution of subscribers across all member links.

**Locally pinned**: Refers to subscribers pinned to a member links of the LAG of the Active node.

**Remotely pined**: Refers to subscribers pinned to a member links of the LAG of the Standby node.

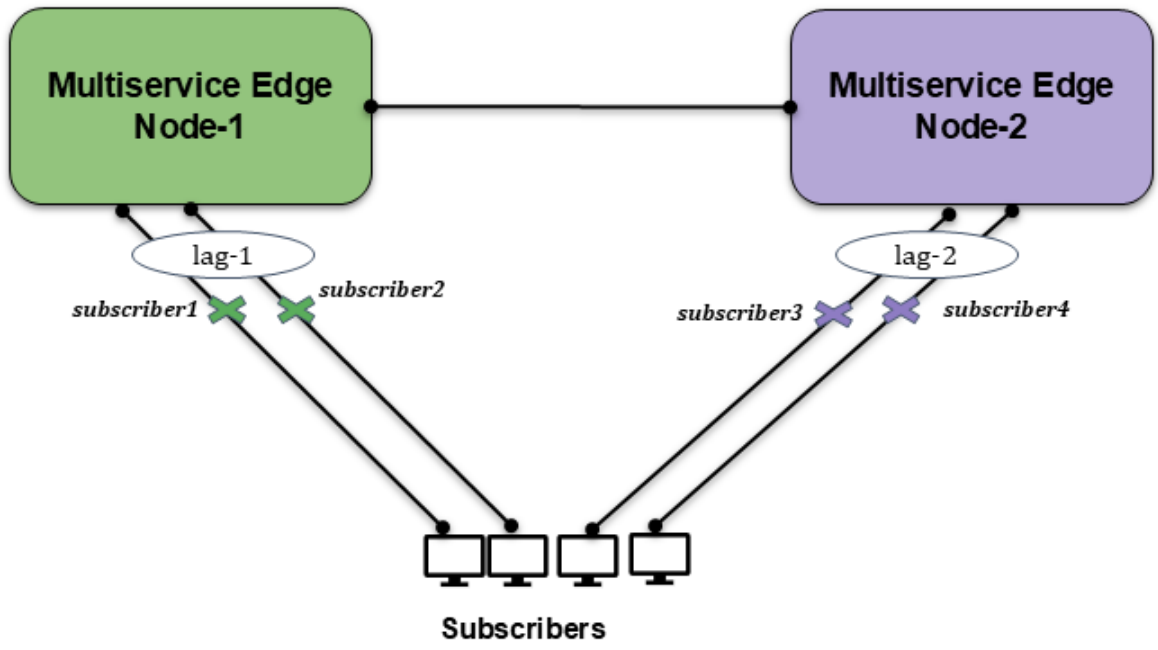



*Figure 100. Pinning a Subscriber to a Link of a LAG*

OLT is excluded from the diagram to simplify it and is not directly relevant to what is depicted.

The diagram illustrates the pinning mechanism, where Subscriber 1 is pinned to Link 1 of LAG 1, and Subscriber 2 is pinned to Link 2 of LAG 1. Similarly, Subscriber 3 is pinned to Link 1 of LAG 2, while Subscriber 4 is pinned to Link 2 of LAG 2.

> When a member link of the access LAG fails on either the active or standby device in the Active-Active Redundancy mode, the system retriggers the subscriber pinning process to maintain an equal distribution of subscribers across the access LAG member links.

- Locally pinned subscribers will be in 'established' state on the active node and in the standby state on the Standby node.
- Remotely pinned subscribers will be in the 'established non-forwarding' state on the active node and in the 'standby-forwarding' state on the standby node.

**Subscriber Pinned to Active Node**

The following table provides the various subscriber state for the Active-Active Redundancy.

| Subscriber Pinning | State on Active Node | State on Standby Node |
|---|---|---|
| Pinned to Active Node | Established | Standby |
| Pinned to Standby Node | Established Non-Forwarding | Standby-Forwarding |

The following example shows the IPoE subscribers pinned to different LAGs and their states.

```
supervisor@rtbrick.net: cfg> show subscriber
```

```
Subscriber-Id          Interface         VLAN       Type    State
1369375761697341460    lag-1             1001:1001  IPoE    ESTABLISHED
1369375761697341463    lag-1             1001:1004  IPoE    ESTABLISHED_NO_FWD
2522297266304188429    lag-11            2001:1001  IPoE    STANDBY
2522297266304188430    lag-11            2001:1002  IPoE    STANDBY_FWD
```

The subscriber ID `1369375761697341460` is pinned to 'lag-1' and is in established state on Active node. The subscriber ID `1369375761697341463` is pinned to 'lag-1' is in a established-non-forwarding mode on the active node. The subscriber ID `2522297266304188429` is pinned to 'lag-11' and is in standby mode on the Standby node. The subscriber ID `2522297266304188430` is pinned to 'lag-11' and is in standby-forwarding mode on the Standby node.

#### Active-Standby Redundancy

In the active-standby node cluster, the active node (for a redundancy session) performs subscriber services. The standby device mirrors concurrent subscriber state data from the active peer (for that redundancy session). Both the nodes, paired for redundancy, keep sending 'keepalive' messages to each other to check the health status.

In this mode, the standby RBFS node does not forward traffic while the active RBFS node is functioning. The standby node stays synchronized with the active node by maintaining up-to-date routing tables, configurations, and session data, but it does not perform any traffic forwarding. If the active node fails, the standby node will take over its role and start forwarding traffic.

Active node handles all subscriber traffic and control plane processing. In this model, the standby LAG is not utilized for traffic unless a failover occurs.

When a LAG is in standby mode with the `standby-non-forwarding` profile enabled, the LAG interface is changed to operationally down state.

### LACP Out-of-Sync Support for Standby LAGs

When a redundancy LAG is configured with the `standby-not-forwarding` mode, LACP packets are transmitted with the LACP state set to Out of Sync (OOS). This prevents remote LACP timer changes and ensures accurate redundancy state resolution.

#### IP Address Pool Management for Redundancy

There is no synchronization between local IP address pools, so an IP address on one BNG will not be reflected in the other BNG's IP address pool.

So, it is highly recommended to use RADIUS for IP address management. If local pools are used, they must be configured with disjoint address ranges across all nodes. While an IP address assigned by a pool on the active BNG can be transferred to the standby BNG during failover, the local pools are not synchronized.

#### Subscriber Services on Redundancy

Subscriber services refer to the various features that can be applied to subscribers, such as access control lists, HTTP redirect, lawful interception, CGNAT, IGMP, and Accounting. When deploying

these services in a redundancy setup, special handling is required. The behavior of each service varies depending on whether the Redundancy mode is active-active or active-standby.

The following table provides information about the various services supported on Active-Active Redundancy and Active-Standby Redundancy.

| Service | Active-Active | Active-Standby |
|---|---|---|
| Accounting | Supported: Both BNGs perform accounting and RADIUS aggregates counters. | Supported: Only the active BNG performs accounting. |
| Lawful Interception | Supported: LI must be enabled on both BNGs. | Supported: LI happens in active BNG and the data is synced to standby BNG. |
| HTTP Redirect | Supported: Enabled on both BNGs. | Supported: Enabled only in active BNG and synced to standby BNG. |
| CGNAT | Unsupported | Unsupported |
| Subscriber ACL | Supported: ACLs are programmed on both BNGs. | Supported: ACLs programmed only in active BNG and synced to standby BNG. |
| ADF (Application Detection and Filtering) | Unsupported | Unsupported |
| Dynamic QoS Profiles | Supported: QoS profile changes via CoA sync on both BNGs. | Supported: QoS profile changes via CoA sync to standby BNG. |
| Dynamic QoS Parameters | Unsupported | Unsupported |
| IGMP (Multicast Services) | Unsupported | Unsupported |

**Inter-BNG Connectivity**

Multiservice Edge platforms, deployed as redundant pairs, are connected through a Redundancy session to maintain state consistency. This Redundancy session uses a TCP connection between the active and standby BNG nodes over an IP/MPLS link. TCP connection can be formed using the directly connected interfaces on the two Multiservice Edge nodes through an IP/MPLS path. The session between the Multiservice Edge nodes is used to send 'keepalive' messages and mirror data in real time. Both the active and standby BNG nodes advertise subscriber routes using the same metric.

RBFS Redundancy mechanism provides several advantages. Downstream traffic arrives either on the active or standby BNG since both advertise subscriber routes with the same metric. If the traffic arrives on the active BNG, it is delivered directly to the subscriber. If the traffic arrives at the standby BNG, it redirects the traffic to the active BNG over the inter-BNG connection.

Another advantage is that the Multiservice Edge nodes do not withdraw or re-advertise subscriber routes if an access link goes down. A link-down event on the access side does not trigger any change in route advertisements on the core side. When an access link fails, the traffic redirection from the

impacted BNG to the other (redundant) BNG occurs locally and quickly without re-advertising or withdrawing routes in the core network.

In the event of a redundancy session failure, 'keepalive' messages can no longer be exchanged between the peer nodes. After the hold timer expires, the affected node transitions to a standalone operational state. All previously synchronized subscriber session data between the nodes becomes invalid.

#### IPoE RD-based Stateful Redundancy Architecture

The following architectural diagram provides a high-level view of RD-based IPoE stateful redundancy mode. It shows two RBFS nodes, paired for redundancy, deployed in an active-standby node cluster, with their interfaces are connected with an RD TCP connection. These peer nodes use the RD TCP connection for sending 'keepalive' messages and data mirroring for subscriber state synchronization.

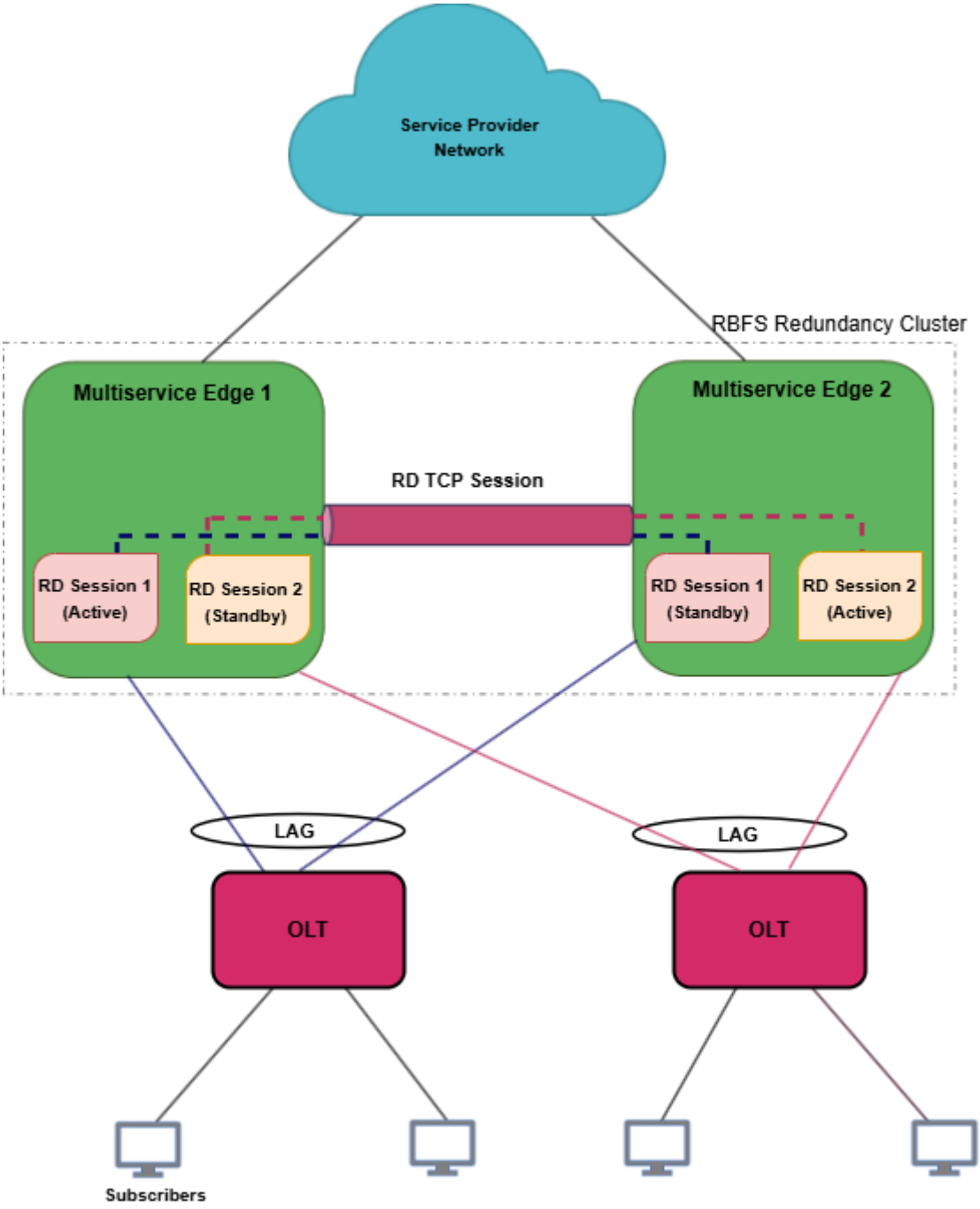



*Figure 101. IPoE RD-based Stateful Redundancy Architecture*

Both of the nodes are connected to an access device (OLT device in this scenario) on one end from where it receives subscriber traffic and sends traffic. The nodes are also connected to the core network on the other end.

The node 'Multiservice Edge 1' is in active state and performs subscriber services for the 'Session 1' (redundancy subscriber group). 'Session 1' is also mirrored in the Multiservice Edge 2 in standby mode. The standby device mirrors concurrent subscriber state data for 'Session 1' from the active peer. If an active node goes down due to any reason, the peer node detects the outage and uses mirrored 'Session 1' to perform subscriber services.

One Multiservice Edge node acts as an active node for one or more sessions (subscriber redundancy groups) and as a standby Multiservice Edge for other subscriber redundancy groups at the same time. The following diagram illustrates the scenario.

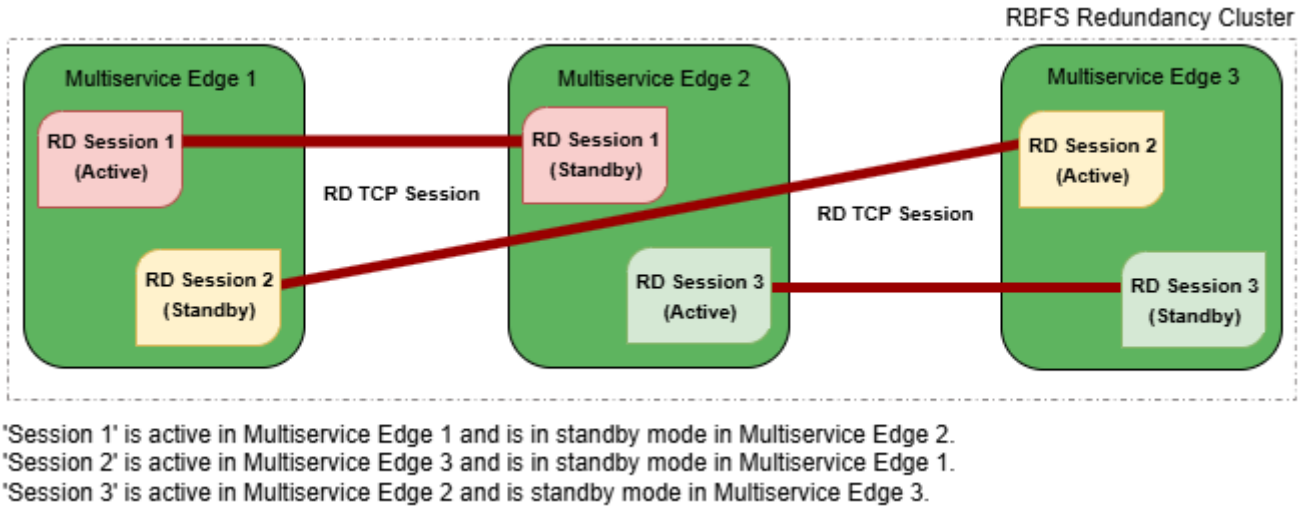



*Figure 102. IPoE RD-based Stateful Redundancy*

IPoE RD-based Stateful Redundancy can mitigate the following types of failures:

- Link failure Between Active RBFS Node and Access Node (OLT, DSLAM or MSAN)
- Node Outage

**Redundancy for Node Outage**

A node outage, which can bring down the subscriber services, can occur due to many reasons on a network. IPoE RD-based Stateful Redundancy helps to minimize the impact and reduce interruptions and downtime by providing a resilient system. In the event of a node outage, IPoE RD-based Stateful Redundancy triggers switchover in which the standby node takes over from the active node with very minimal impact on the subscriber services.

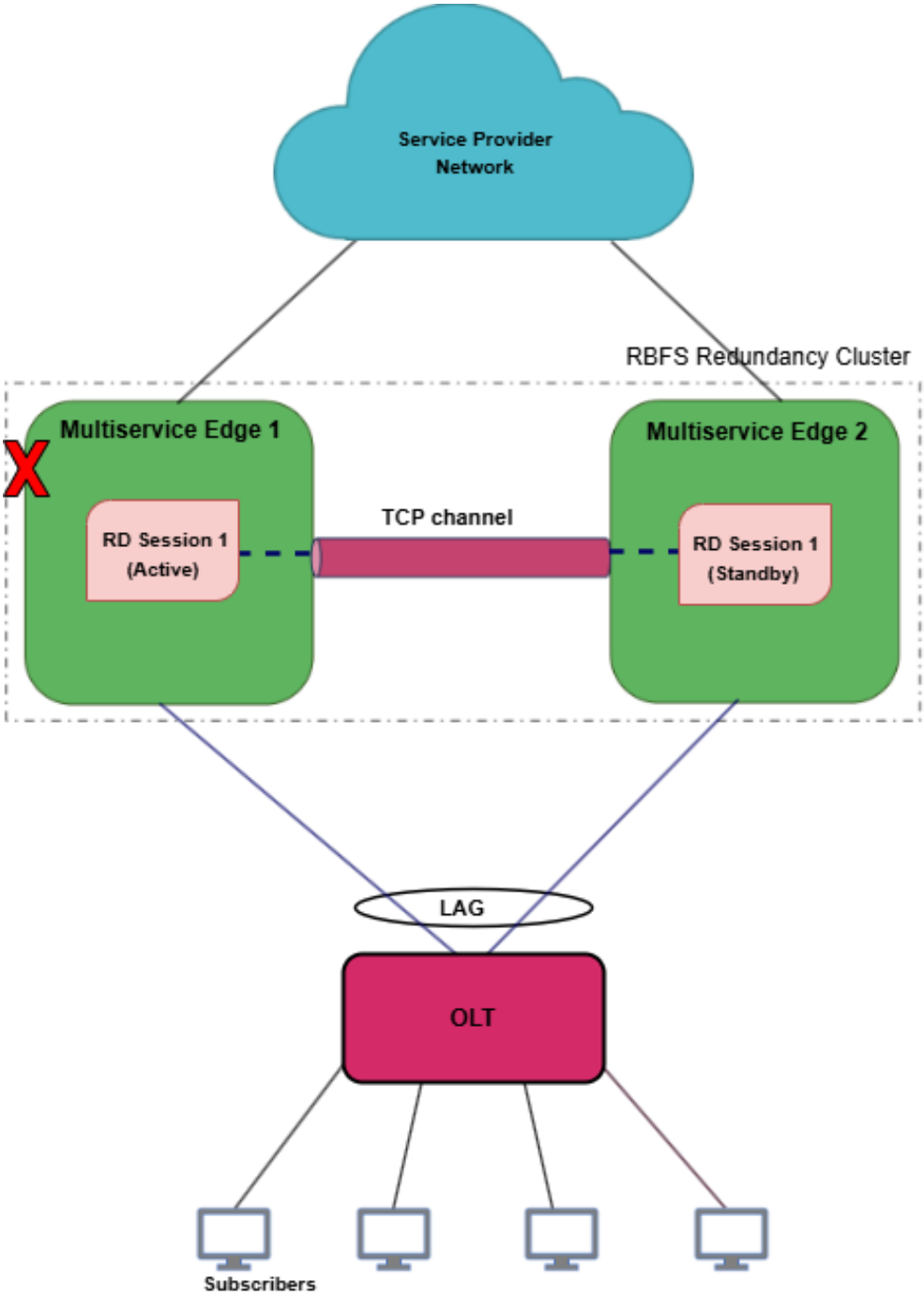



*Figure 103. Node Outage Scenario*

The diagram shows `multiservice-edge1` as an active node serving subscribers and `multiservice-edge2` stays as a standby. When an active node goes down, the standby node detects the same and takes over from the active RBFS node. In a node outage scenario, the node becomes unresponsive and cannot maintain communication with its peer node the RD TCP link.

**Redundancy for Link Failure**

In IPoE RD-based Stateful Redundancy, Link Aggregation combines multiple physical links into a single logical link. If a member link, which is part of a LAG, goes down, the LAG fails. The diagram shows the 'multiservice-edge1' and 'multiservice-edge2' deployed in redundancy mode and is connected to the access device with LAGs. When the LAG between 'multiservice-edge1' and the access node goes down, the 'multiservice-edge1', which is running 'Session 1', becomes inactive for the subscriber group. So the 'multiservice-edge2', which is the standby for 'Session 1', detects the

failure of 'multiservice-edge1' and starts performing subscriber services for 'Session 1' by providing a quick recovery from the disruption.

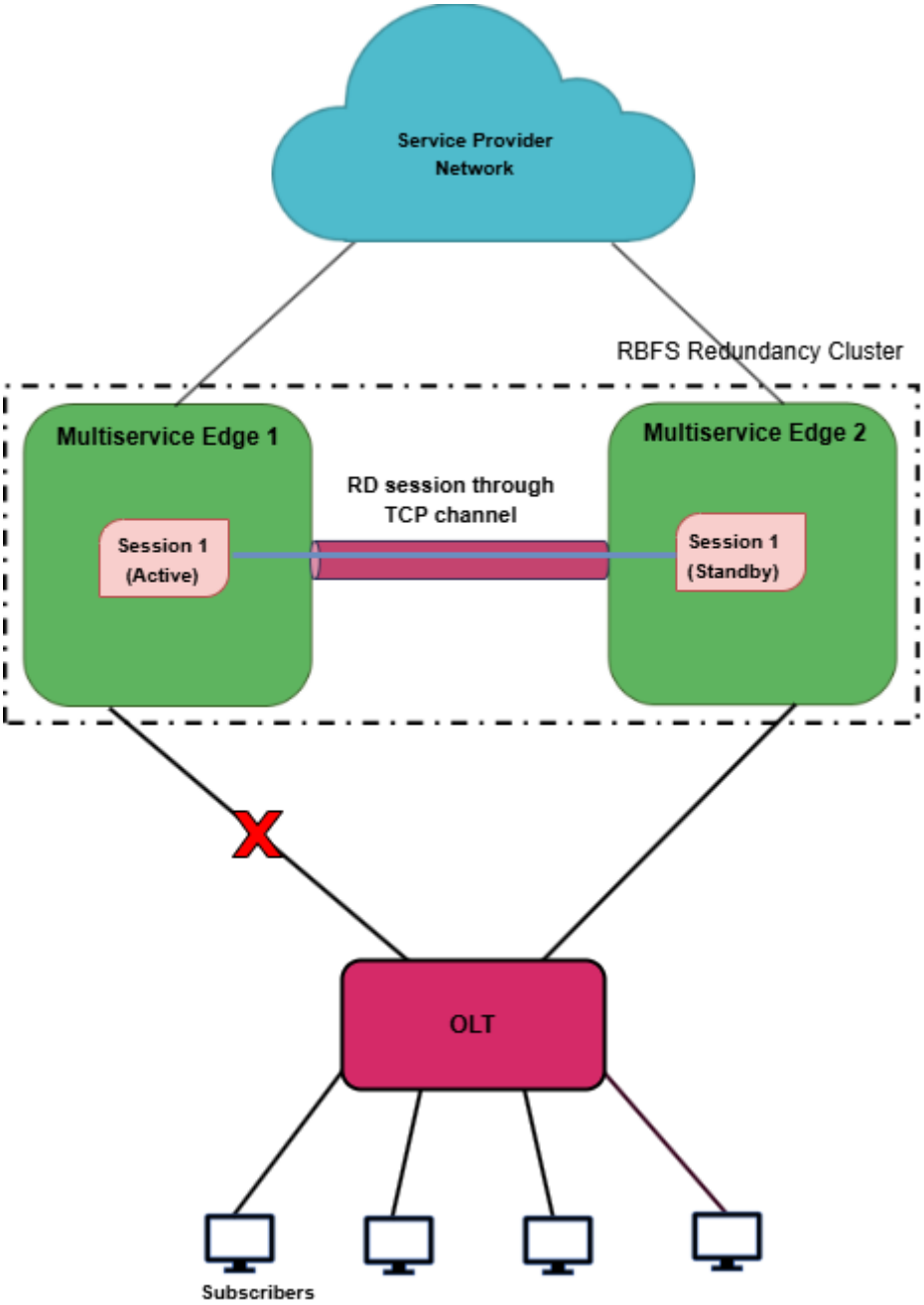



*Figure 104. Link Failure Scenario*

In this link or LAG failure scenario, `multiservice-edge1` is in a healthy state, only the LAG interface went down. It cannot perform subscriber services for the particular redundancy session. However, it can keep communication with its peer node as the RD TCP channel.

#### Node States in Redundancy

In IPoE RD-based Stateful redundancy, Multiservice Edge nodes have different states for various redundancy sessions. Typically, IPoE Stateful redundancy nodes encounter the following states for redundancy sessions:

**Active:** All subscribers are served by active node in the RBFS active-standby node cluster. One node which is active for a redundancy session can be a standby node for a different session. Nodes, that

are paired for redundancy, send 'keepalive' messages to each other and also synchronize all subscriber state data with the peer node. The priority values that you specify for the redundancy nodes determine the roles of active and standby. The node that receives the higher priority value for the session ID assumes the role of active for that subscriber group. To set one device as 'active', you must specify higher priority value for the redundancy session for that node.

**Standby:** Standby node is identical with the active node and synchronizes subscriber data concurrently from peer node. It keeps communication with the peer node to monitor node health status using 'keepalive' messages. The node that gets the lower priority value for the redundancy session ID assumes the role of standby. Standby node for a subscriber group does not perform any subscriber services for that group unless or until the active node encounters an outage.

**Down**: When a node becomes inactive due to an outage, it is considered as 'down'. In the event of a node outage, it is completely down and cannot perform subscriber services and any communication with its peer node. But in the case of a LAG (between the node and access node) failure, the node cannot perform subscriber services, but it can communicate with the peer node through the RD TCP connection. So that the subscriber state synchronization occurs without any interruption.

**Stand Alone:** When the active node goes down, the switchover occurs and standby takes over the subscriber service. In this scenario, the serving node is in 'stand alone' state (for that redundancy session) as it has no peer node for redundancy.

#### Revert or Rollback Active-Standby Redundancy Mode

In IPoE RD-based Stateful Redundancy, after a node or link failure and the subsequent switchover, the standby takes over and continues the subscriber service for that subscriber group even after the other node (previously active) recovers from the failure. There is no automated rollback or revert to the previously active router. However, administrators can perform a manual switchover for the Active-Standby Redundancy mode.

#### Monitoring Node Health Status

The RBFS nodes, which have switchover capacities, monitor each other for the health status. IPoE RD-based Stateful Redundancy uses 'keepalive' messages that check on the health of the RBFS nodes. Both of the devices send 'keepalive' messages to each other in every five seconds. One node can detect a failure if it does not receive 'keepalive' messages for a period of 20 seconds from the other node.

#### Redundancy Clients

There are multiple RBFS daemons that participate in providing redundancy. They include redundancy daemon, (`rd`), LAG daemon (`lagd`), interface daemon (`ifmd`), subscriber daemon (`subscriberd`), IPoE daemon (`ipoed`) and forwarding daemon (`fibd`). These daemons, which perform various roles, are known as redundancy clients.

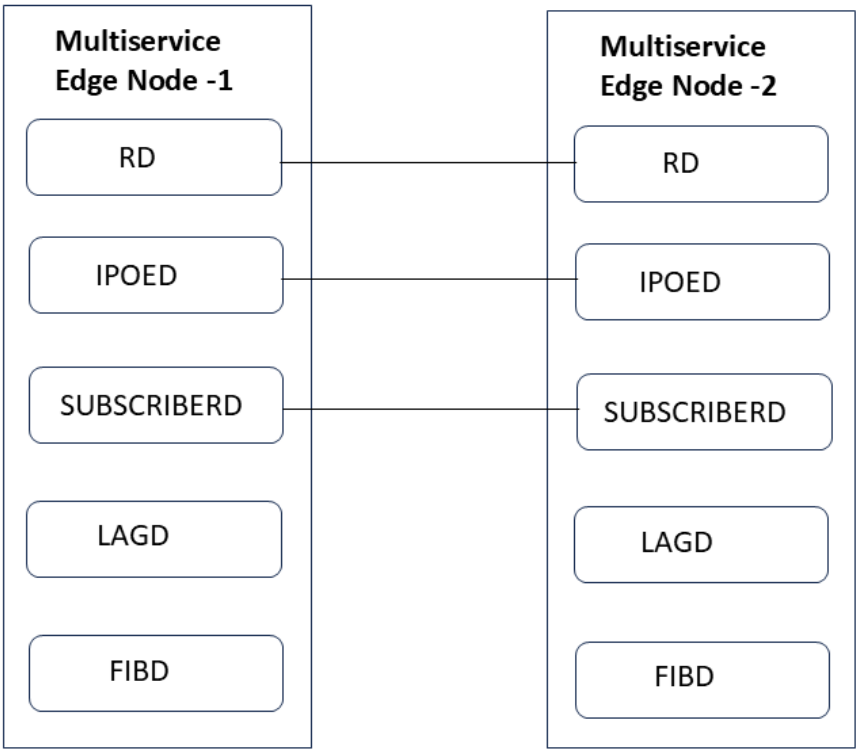



*Figure 105. Redundancy Clients*

**Redundancy in DHCP Relay Mode**

RBFS supports RD-based Stateful Redundancy for IPoE in both DHCPv4 and DHCPv6 relay modes. RBFS can maintain synchronized IPoE session states across redundant Multiservice Edge nodes operating in 'relay' mode.

> In DHCP Relay mode, source address configured for DHCP server on both of the Multiservice Edge nodes should be different, and the proxy-ARP "any" should be enabled.

**Redundancy Daemon**

Redundancy Daemon is responsible for establishing high-availability connections. It monitors the ecosystem and detects any outages that may happen on the network. It assigns the roles of active or standby to the nodes depending on the priority configured on the node. The daemon triggers a switchover to the standby node if a failure occurs. It responds to the failure events which are reported locally by daemons who are the redundancy clients. It also reconciles the data after switchover from the node that went down.

### Supported Hardware Platforms

Currently, RBFS can be deployed in redundancy mode using the RBFS Multiservice Edge switches. Multiservice Edge software provides complete BNG functionalities on a single compact hardware switch. Both the Active-Active Redundancy and Active-Standby Redundancy modes are supported in the following hardware platforms:

- UfiSpace S9600-72XC: The UfiSpace S9600-72XC is a multi-function, disaggregated white box aggregation routing platform that is equipped with Broadcom’s Qumran2c chipset. It features

64x25GE and 8x100GE high-speed ports with a switching capacity of up to 2.4Tbs.

- Edgecore AGR420: AGR420 is a high performance 25GbE aggregation router that consists of fixed 64 x 10G/25G SFP28, 8 x 100GE QSFP28 and 2 x 100G QSFP-DD network interface configurations.

Only Active-Standby Mode is supported on the following platforms:

- Edgecore CSR440
- UfiSpace S9510-28DC

**IPoE RD-Based Stateful Redundancy Requirements**

The following are the requirements for setting up Redundancy.

- Ensure that both of the platform devices, on which RBFS software runs, must be the same model.
- Ensure that the devices should run the same version of RBFS software. RBFS software 23.2.1 and later versions support Active-Standby redundancy mode and 25.1.1 and later versions support Active-Active Redundancy mode.
- NTP must be configured on both devices to match the timestamps.

## Configuration

### Redundancy

This section provides information on how to deploy Multiservice Edge device pair to achieve redundancy. The following workflow diagram depicts the end-to-end deployment of Multiservice Edge in Redundancy mode:

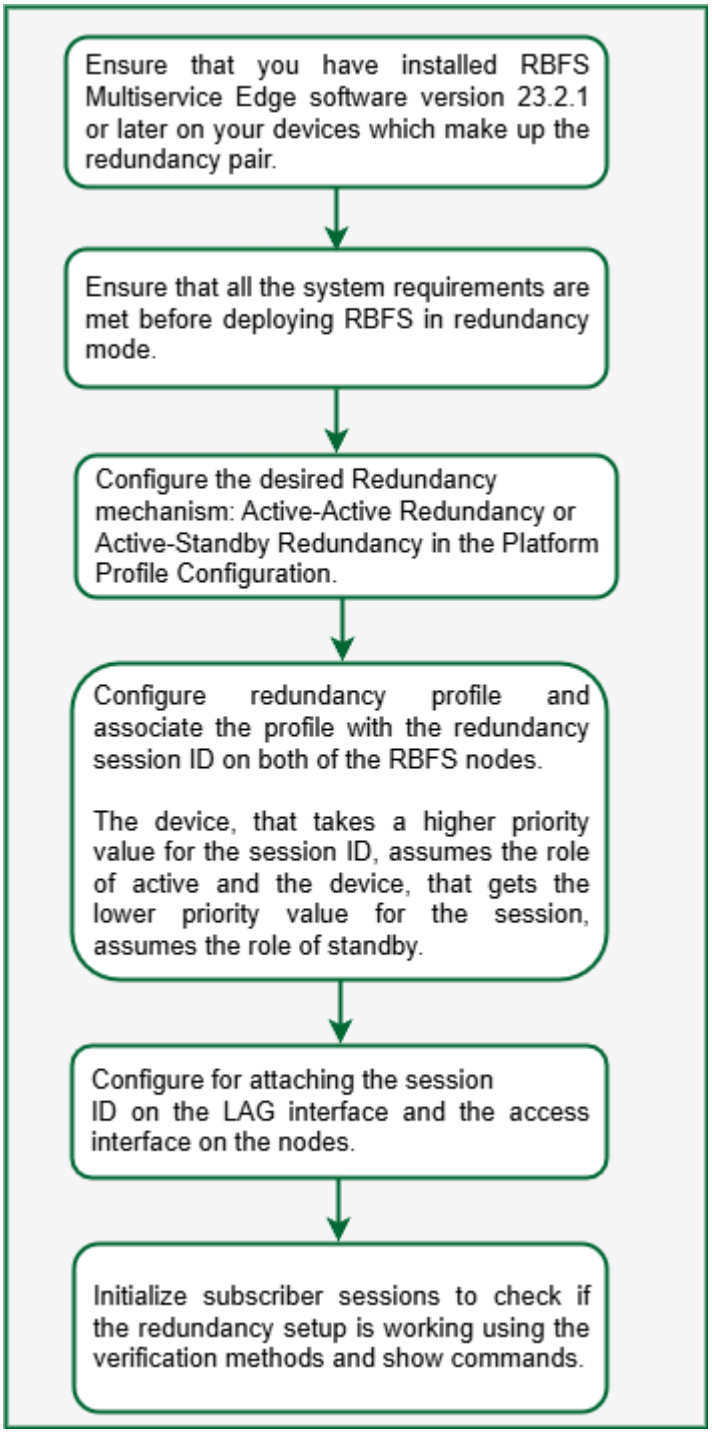



*Figure 106. Workflow for Multiservice Edge in Redundancy Mode*

You must perform the following tasks to deploy the RBFS devices in redundancy mode. These configurations must be performed on both of the devices.

1. Set up the Redundancy Mode: `Standby Forwarding` or `Standby Non-Forwarding`.
2. Configure Redundancy Profile
3. Configure Session for Redundancy
4. Configure Link Aggregation Group for Redundancy
5. Configure Access for Redundancy

#### Configuration Syntax and Commands

The following sections describe syntax and commands for various configurations.

#### Setting up the Redundancy Mode

You can choose your preferred redundancy mechanism and set up the system to enable it. Redundancy is available in two modes: `Standby Forwarding` and `Standby Non-Forwarding`. By default, the `Standby Forwarding` method is enabled. If you would like to use the `Standby Non-Forwarding` mode instead, you need to select the `Standby Non-Forwarding` option in the Platform Profile configuration.

> This setup is a one-time activity and system reboot is required the changes to take effect.

To set the desired redundancy mode, use the command.

**Syntax:**

**set system platform profile** <profile-name> **redundancy-mode** [**Standby-Forwarding** | **Standby-Non-Forwarding**]

| Attribute | Description |
|---|---|
| `standby-forwarding` | Active-Active redundancy mode. Redundancy mode in which the standby LAG is utilized for traffic forwarding. |
| `standby-non-forwarding` | Active-Standby Redundancy mode. Redundancy mode in which Standby node LAG does not forward traffic unless a failover occurs. |

Example:

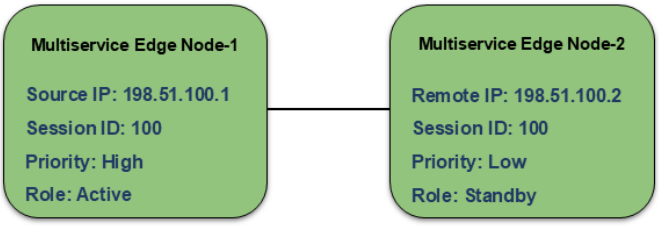
```
supervisor@rtbrick.net: cfg> show config system platform profile
{
  "rtbrick-config:profile": [
    {
      "profile_name": "nat_4q",
      "redundancy-mode": "standby-non-forwarding"
    }
  ]
}
```

For more information about Platform Profile, see section Platform Profile Support.

#### Configuring Redundancy Profile

Redundancy profile configuration is used to provide peer identity in redundancy. While configuring the redundancy profile on the nodes, you need to specify IP addresses of both the peer nodes. Based on the priority value that you specify for the session ID, the peers take the roles of

active and standby for the Session.




**Syntax:**

**set redundancy profile** <name>

| Attribute | Description |
|---|---|
| peer | Redundancy configuration |
| switchover-hold-timer | Minimum time interval between consecutive switch-overs in seconds. Default is 60 seconds. Range: [0- 600]. |

Run the following commands to configure redundancy profile.

```
set redundancy profile rd_ipoe
set redundancy profile rd_ipoe peer ipv4 remote-address 198.51.100.2
set redundancy profile rd_ipoe peer ipv4 update-source 198.51.100.1
set redundancy profile rd_ipoe peer ipv4 instance default
```

Example Configuration:

```
supervisor@rtbrick>cbng1.rtbrick.net: cfg> show config redundancy profile
{
    "rtbrick-config:profile": [
      {
        "name": "rd_ipoe",
        "peer": {
          "ipv4": {
            "remote-address": "198.51.100.2",
            "update-source": "198.51.100.1",
            "instance": "default"
          }
        }
      }
    ]
  }
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Configuring Session for Redundancy

You can configure Redundancy Session with a unique session ID. You can define the system priority value (which determines active and standby roles) and the associate Redundancy profile configuration in the Redundancy Session configuration.

**Syntax:**

**set redundancy session** <session-id>

| Attribute | Description |
|---|---|
| `keepalive-interval` | Keepalive message transmission interval in seconds. Default is 5 seconds. |
| `priority` | Session priority. Allowed Range: 0 - 65535. |
| `profile` | Profile name |

Run the following commands to configure session for redundancy.

```
set redundancy session 100
set redundancy session 100 priority 10
set redundancy session 100 profile rd_ipoe
```

Configuration Example:

```
supervisor@rtbrick>cbng2.rtbrick.net: cfg> show config redundancy session
{
    "rtbrick-config:session": [
      {
        "session-id": 100,
        "priority": 10,
        "profile": "rd_ipoe"
      }
    ]
  }
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Configuring LAG for Redundancy

You must associate the Redundancy Session with the LAG. While configuring LAG for redundancy, you must specify the session ID to associate the LAG with the Redundancy Session. LAG can identify its Redundancy Session with this mapping.

**Syntax:**

**set link-aggregation interface** <interface-name> **options**

| Attribute | Description |
|---|---|
| `description` | Link aggregation interface description. |
| `member-interface` | Link aggregation member interface configuration |
| `minimum-link-count` | Minimum number of active member links required for the link aggregation interface. default values is 1. |
| `mode` | Mode of the link aggregation interface, static or lacp. default mode is lacp. |
| `rd-role` | Role of the link aggregation interface, active or standby. |
| `rd-system-priority` | The value for the system priority range from 1 to 65535. The lower the value, the higher the system priority. default value is 65535. |
| `redundancy-session-id` | The value for the redundancy group session id range from 1 to 65535. |
| `system-id` | Redundancy System ID of link-aggregation interface. |

Run the following commands to configure LAG for redundancy.

```
set link-aggregation interface lag-1
set link-aggregation interface lag-1 mode lacp
set link-aggregation interface lag-1 minimum-link-count 1
set link-aggregation interface lag-1 redundancy-session-id 100
set link-aggregation interface lag-1 system-id a8:b5:7e:8f:66:43
set link-aggregation interface lag-1 member-interface ifp-0/1/260
```

Example Configuration:

```
supervisor@rtbrick>cbng2.rtbrick.net:: cfg> show config link-aggregation interface
lag-1
{
    "rtbrick-config:interface": [
      {
        "interface-name": "lag-1",
        "mode": "lacp",
        "minimum-link-count": 1,
        "redundancy-session-id": 100,
        "system-id": "11:22:33:44:55:66",
        "member-interface": [
          {
            "member-interface-name": "ifp-0/0/4"
          }
        ]
```

```
      }
    ]
  }
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Configuring Access for Redundancy

IPoE and Redundancy Session mapping is essential to associate the Redundancy Session with IPoE. While configuring access for redundancy, you must specify access type as IPoE.

**Syntax:**

**set access interface double-tagged** <name> <options>

| Attribute | Description |
|---|---|
| aaa-profile-name | AAA profile name |
| access-profile-name | Access profile name |
| access-type | Access service type |
| gateway-ifl | IPoE gateway IFL (unnumbered source IFL) |
| max-subscribers-per-mac | Restrict maximum subscribers per MAC address |
| max-subscribers-per-vlan | Restrict maximum subscribers per VLAN |
| redundancy-session-id | Redundancy session id for this interface |
| service-profile-name | Service profile name |
| vlan-profile-enable | Enable VLAN profiles |

Run the following commands access for redundancy.

```
set access interface double-tagged lag-1 1001 1100 1001 1100
set access interface double-tagged lag-1 1001 1100 1001 1100 access-type IPoE
set access interface double-tagged lag-1 1001 1100 1001 1100 access-profile-name ipoe
set access interface double-tagged lag-1 1001 1100 1001 1100 aaa-profile-name ipoe-aaa
set access interface double-tagged lag-1 1001 1100 1001 1100 gateway-ifl lo-0/0/0/10
set access interface double-tagged lag-1 1001 1100 1001 1100 redundancy-session-id 100
```

Example Configuration:

```
supervisor@rtbrick>cbng1.rtbrick.net: cfg> show config access interface double-tagged
lag-1
{
    "rtbrick-config:double-tagged": [
      {
        "interface-name": "lag-1",
```

```
        "outer-vlan-min": 1001,
        "outer-vlan-max": 1100,
        "inner-vlan-min": 1001,
        "inner-vlan-max": 1100,
        "access-type": "IPoE",
        "access-profile-name": "ipoe",
        "aaa-profile-name": "ipoe-aaa",
        "gateway-ifl": "lo-0/0/0/10",
        "redundancy-session-id": 100
      }
    ]
  }
```

To access the RESTCONF API that corresponds to this CLI, click here.

#### Switchover Manually

In the Active-Standby redundancy mode, administrators can perform a manual switchover from an active node to a standby node. Use the following command to perform switchover:

**Syntax:**

```
*request switch-over session* <session-id> *confirm*
```

Example:

```
supervisor@rtbrick>: cfg> request switch-over session 1 confirm
Switchover has started!
```

### Operational Commands

#### Redundancy Show Commands

With the RBFS Command Line Interface, you can view output of operational commands. The redundancy operational commands provide detailed information about the RBFS redundancy operations.

#### Redundancy Details

**Syntax:**

**show redundancy** <options>

| Option | Description |
|---|---|
| client | Displays information about redundancy client daemons. |
| session | Displays session details. |

#### Client Statistics

**Syntax:**

**show redundancy client** <name> **statistics**

This command displays information from redundancy client (daemon) which is a participant in the redundancy sessions. The daemons `ifmd`, `ipoed.1`, `lagd`, and `subscriberd.1` are the client daemons in redundancy sessions.

Command:

```
show redundancy client lagd statistics
```

```
supervisor@rtbrick: op> show redundancy client lagd statistics
Session id: 100, Profile: IPoE_OLT
  Agent State: active
  TCP operational state: up
  Message statistics:
    Keepalive sent    : 0
    Keepalive received: 0
  Last 5 state changes: [Latest first]
    active              : 2026-06-05T17:12:24.532558+0000
    promote-ready       : 2026-06-05T17:12:24.532485+0000
    promote-infra-wait  : 2026-06-05T17:12:24.532414+0000
    promote-app-wait    : 2026-06-05T17:12:24.530439+0000
    down                : 2026-06-05T17:12:14.492134+0000
  Connection statistics:
    Peer Address: 192.0.2.6
    Application down notifications: 0
    Connection down notifications: 0
    Retry count: 0
    Session down received: 0
```

#### Redundancy Session Details

**Syntax:**

**show redundancy session** <id> <options>

| Option | Description |
|---|---|
| `-` | Without any option, it displays all session details. |
| `detail` | Displays details session information. |
| `history` | Displays session history. |

| Option | Description |
| --- | --- |
| status | Displays session status. |

Example: Redundancy session details

```
supervisor@rtbrick: op> show redundancy session detail
Redundancy session ID: 100, Source: 192.0.2.5, Peer: 192.0.2.6
  Instance: default, Profile name: IPoE_OLT, Local priority: 20
  State: active, Previous state: connect, Last state transition time: 2026-06-
05T17:12:24.524702+0000
  Peer state: standby
  Mode: standby-forwarding
  Interface: lag-1
  TCP operational state: up
  Message statistics:
    Keepalive sent       : 3
    Keepalive received   : 2
    Keepalive ignored    : 0
  Switchovers via cli    : 0
  State transition statistics:
    Down                 : 0
    Connect              : 1
    Standby              : 0
    Active               : 1
    Stand-alone          : 0
    Demote               : 0
  Timers:
    Connect retry interval              : 4294967295
    Keepalive timer interval            : 5000
    Holddown timer interval             : 20000
    Switchover timer pending interval   : 0 sec
  Timer Statistics:
    Holddown timer expiry count         : 0
```

Example: RD session detail for a session ID.

```
supervisor@rtbrick: op> show redundancy session detail
Redundancy session ID: 100, Source: 192.0.2.5, Peer: 192.0.2.6
  Instance: default, Profile name: IPoE_OLT, Local priority: 20
  State: active, Previous state: connect, Last state transition time: 2026-06-
05T17:12:24.524702+0000
  Peer state: standby
  Mode: standby-forwarding
  Interface: lag-1
  TCP operational state: up
  Message statistics:
    Keepalive sent       : 3
    Keepalive received   : 2
```

```
    Keepalive ignored    : 0
  Switchovers via cli    : 0
  State transition statistics:
    Down                 : 0
    Connect              : 1
    Standby              : 0
    Active               : 1
    Stand-alone          : 0
    Demote               : 0
  Timers:
    Connect retry interval              : 4294967295
    Keepalive timer interval            : 5000
    Holddown timer interval             : 20000
    Switchover timer pending interval   : 0 sec
  Timer Statistics:
    Holddown timer expiry count         : 0
```

Example: Status information of a session specified

```
supervisor@rtbrick: op> show redundancy session 100 status
State: active,  Remote State: standby
Redundancy client replication information:
  Total redundancy clients : 4
  ifmd:
  ipoed.1:
    Number of subscribed table: 1
    Number of published table: 1
    Number of table subscription requested: 1
    Table Name                                    Objects
    rd.100.ipoed.packet.in.response               0
  lagd:
  subscriberd.1:
```

Example: History of a session ID for a specified count.

```
supervisor@rtbrick: op> show redundancy session 100 history count 3
Previous state       Current state        State change reason  Timestamp
connect              active               active               2026-06-
05T17:12:24.524702+0000
down                 connect              open                 2026-06-
05T17:12:14.486952+0000
invalid              down                 init                 2026-06-
05T17:12:14.484858+0000
```

Example for detailed output of redundancy session.

```
supervisor@rtbrick>: cfg> show redundancy session detail
```

```
Redundancy session ID: 1, Source: 10.1.1.1, Peer: 10.1.1.2
  Instance: default, Profile name: test, Local priority: 20
  State: down, Previous state: active, Last state transition time: 2026-02-
10T08:27:10.793854+0000
  Peer state: invalid
  Mode: standby-non-forwarding
  Interface: lag-1
  TCP operational state: down
  Message statistics:
    Keepalive sent       : 4412
    Keepalive received   : 4149
    Keepalive ignored    : 0
  Switchovers via cli    : 11
  State transition statistics:
    Down                 : 1
    Connect              : 1
    Standby              : 16
    Active               : 6
    Stand-alone          : 16
    Demote               : 5
  Timers:
    Connect retry interval              : 2000
    Keepalive timer interval            : 5000
    Holddown timer interval             : 20000
    Switchover timer pending interval   : 0 sec
  Timer Statistics:
    Holddown timer expiry count         : 10
```

Example for detailed output of a specified redundancy session.

```
supervisor@rtbrick>: cfg> show redundancy session 1 detail
Redundancy session ID: 1, Source: 10.1.1.1, Peer: 10.1.1.2
  Instance: default, Profile name: test, Local priority: 20
  State: down, Previous state: active, Last state transition time: 2026-02-
10T08:27:10.793854+0000
  Peer state: invalid
  Mode: standby-non-forwarding
  Interface: lag-1
  TCP operational state: down
  Message statistics:
    Keepalive sent       : 4412
    Keepalive received   : 4149
    Keepalive ignored    : 0
  Switchovers via cli    : 11
  State transition statistics:
    Down                 : 1
    Connect              : 1
    Standby              : 16
    Active               : 6
    Stand-alone          : 16
```

```
    Demote              : 5
  Timers:
    Connect retry interval              : 2000
    Keepalive timer interval            : 5000
    Holddown timer interval             : 20000
    Switchover timer pending interval   : 0 sec
  Timer Statistics:
    Holddown timer expiry count         : 10
```

### 5.9.6. Stateless Redundancy for PPPoE Subscribers

#### Overview

RBFS supports RD-based Redundancy for PPPoE subscribers to maintain service continuity during node or link failures. The Redundancy is based on a deterministic active-standby BNG architecture, where only one BNG node actively establishes, maintains and terminates subscriber sessions. The standby BNG node remains idle and ready to take over if there is a failure on the active BNG node.

Subscriber traffic is managed exclusively by the active BNG node and the standby node continuously monitor the health of the active node. PPPoE Redundancy does not synchronize subscriber session state between the active and standby nodes. When a failure occurs, the standby node promotes itself to the active role based on health monitoring mechanism.

When an outage occurs on the active node, all existing PPPoE sessions on that node are terminated. Affected subscribers are required to re-initiate PPPoE discovery and authentication. Subscriber services are restored by establishing new PPPoE sessions on the standby node (standalone) that becomes active. Subscribers, re-onboarded on the new standby node, are assigned a new IP address from its local pool. As a result, downstream traffic is redirected to the new active node without any route advertisements or withdrawals.

- The term 'Multiservice Edge' used in this document refers to an RBFS BNG platform that performs the Multiservice Edge role. The Multiservice Edge role integrates all BNG functionalities within a single platform. In contrast, the spine-leaf architecture distributes these functionalities between spine and leaf platforms based on their respective roles.

#### Understanding RD-Based Stateless Redundancy for PPPoE Subscribers

##### Inter-BNG Connectivity and Redundancy Session

The BNG nodes configured as a redundancy pair maintain a redundancy session over an TCP connection through an IP/MPLS link. TCP connection can be formed using the directly connected interfaces or via an IP/MPLS path through core network. This communication channel is used by the standby node to continuously monitor the health state of the active node. The redundancy session remains established as long as the TCP connection is available. The session between the nodes is used to send 'keepalive' messages to check the node health. In the event of a failure on the active node, the standby node detects the failure through this channel and initiates the failover.

Every redundancy session is represented by a redundancy session ID. When you define a value for

the redundancy session ID, this ID should be unique within the redundancy group or domain and must be configured with the same value on both nodes (active and standby) participating in that redundancy session. When two nodes get the same session ID, they recognize each other as the peer nodes for a particular redundancy session.

From the perspective of the OLT, there is only one active-standby LAG that is distributed across multiple BNGs. This means that the OLT sees one logical link, even though it is physically connected to two separate BNGs.

#### PPPoE Stateless Redundancy Architecture

The following architectural diagram provides a high-level view of PPPoE redundancy mode. It shows two RBFS nodes, paired for redundancy, deployed in an active-standby node cluster, with their interfaces are connected with an TCP connection.

Both nodes are connected to an OLT device on southbound (access) side, through which they receive and forward subscriber traffic. On the northbound side, the nodes are connected to the core network for upstream and downstream communication.

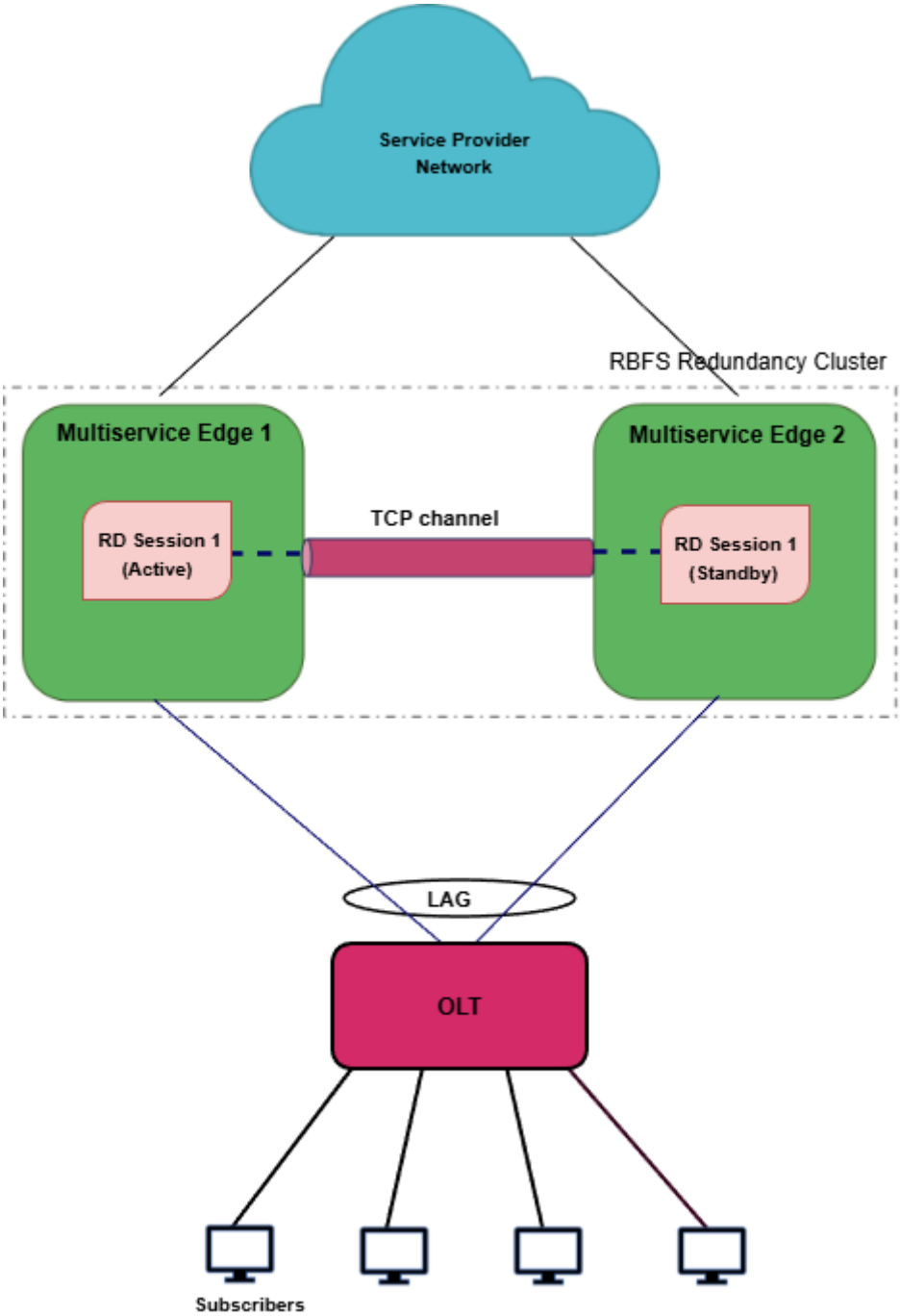



*Figure 107. PPPoE Redundancy Architecture*

PPPoE Redundancy can mitigate the following types of failures:

- Link failure Between Active RBFS Node and Access Node (OLT, DSLAM or MSAN)
- Active Node Outage

**Redundancy for Node Outage**

When the active BNG node fails, the standby node detects the failure through health-state monitoring maintained by the redundancy session. The standby node then transitions to standalone mode, activates the LAG, and re-onboards subscribers. The following diagram illustrates a node outage scenario.

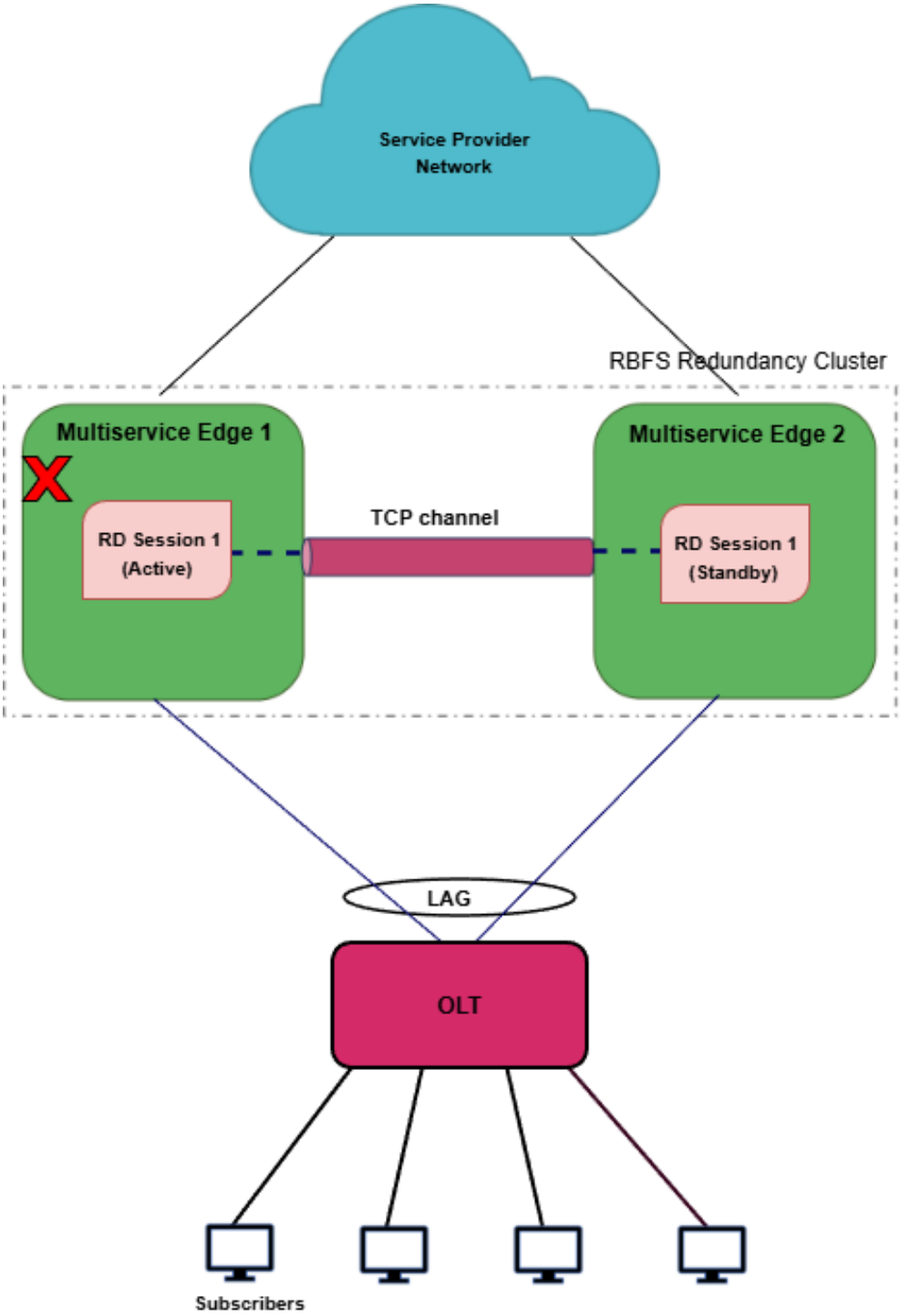



*Figure 108. Node Outage Scenario*

**Redundancy for Link Failure**

A LAG is considered down when the number of member links falls below the configured minimum link threshold. When this occurs, the affected node can no longer serve subscriber traffic. The standby node detects the failure through the TCP connection and initiates failover by re-onboarding subscribers and establishing new sessions. The following illustration depicts a LAG failure scenario in which the active LAG between the 'multiservice-edge1' node and the OLT device goes down.

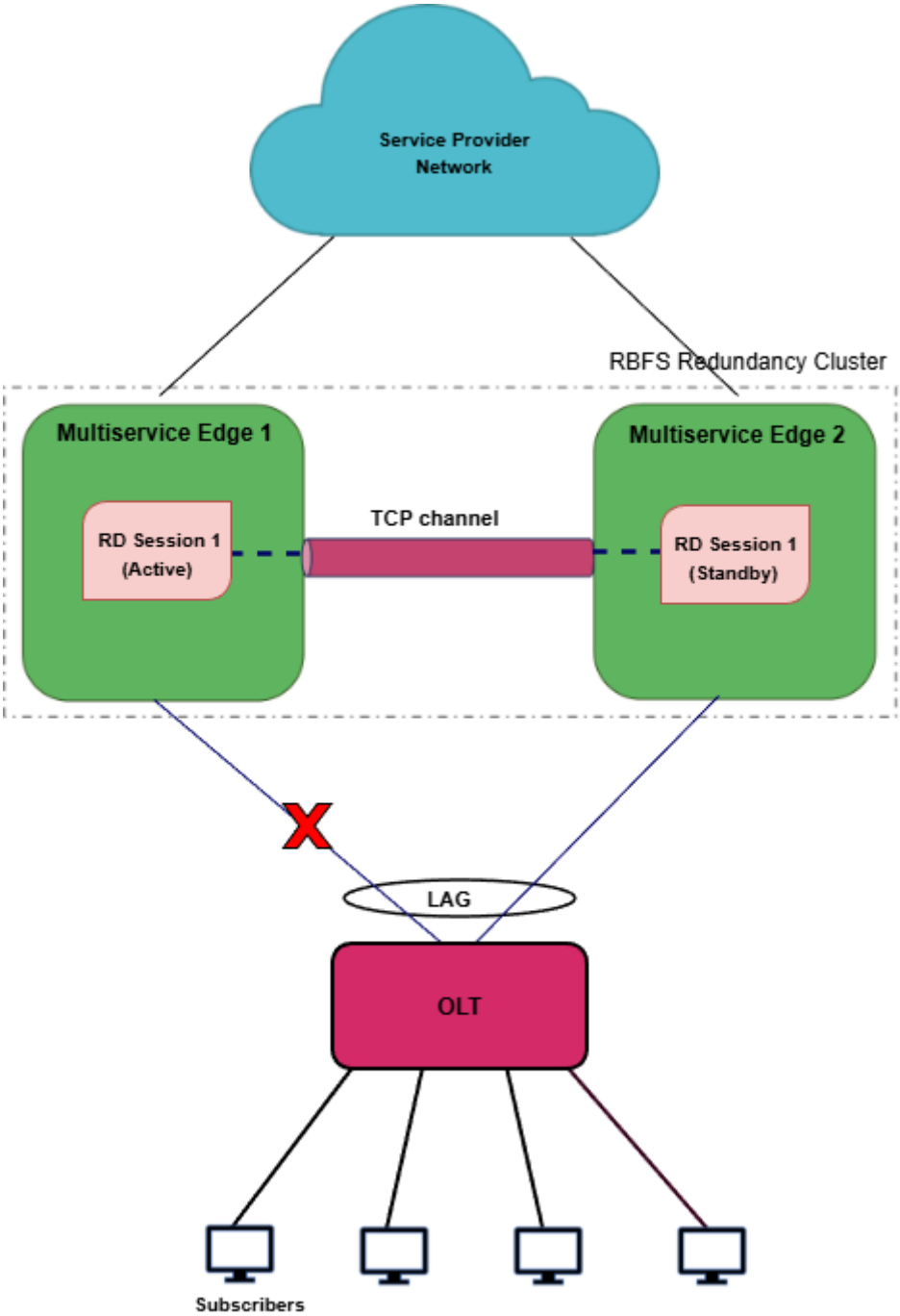



*Figure 109. Link Failure Scenario*

**Node States in PPPoE Redundancy**

Multiservice Edge nodes have different states for various redundancy sessions.

**Active:** All subscribers are served by active node in the RBFS active-standby node cluster. One node which is active for a redundancy session can be a standby node for a different session. The priority values that you specify for the redundancy nodes determine the roles of active and standby. The node that receives the higher priority value for the session ID assumes the role of active for that subscriber group.

**Standby:** Standby node keeps communication with the active BNG node to monitor node health status using the 'keepalive' messages. The node that gets the lower priority value for the redundancy session ID assumes the role of standby. Standby node for a subscriber group does not perform any subscriber services for that group unless or until the active node encounters an

outage.

**Down**: When a node becomes inactive due to an outage, it is considered as 'down'. In the event of a node outage, it is completely down and cannot perform subscriber services and any communication with its peer node. But in the case of a LAG (between the node and access node) failure, the node cannot perform subscriber services, but it can communicate with the peer node through the RD TCP connection.

**Stand Alone:** When the active node goes down, the switchover occurs and standby takes over the subscriber service. In this scenario, the serving node is in 'stand alone' state (for that redundancy session) as it has no peer node for redundancy.

### Supported Hardware Platforms

Currently, RBFS can be deployed in redundancy mode for the PPPoE subscribers using the following RBFS Multiservice Edge platforms.

- UfiSpace S9600-72XC
- UfiSpace S9600-102XC
- UfiSpace S9510-28DC
- Edgecore CSR440 (AS7535-28XB)

### Additional Requirements for PPPoE Redundancy

The following are additional requirements which must be met before setting up Redundancy.

- Ensure that both of the platform devices, on which RBFS software runs, must be the same model.
- Ensure that the devices should run the same version of RBFS software. RBFS software 25.4.1 and later versions support Active-Standby redundancy mode for PPPoE subscribers.
- NTP must be configured on both devices to match the timestamps.

## Configuration

#### Configuration Overview

#### Configuring Redundancy for PPPoE Subscribers

You must perform the following tasks to deploy the RBFS devices in redundancy mode. These configurations must be performed on both of the devices.

1. Set up the Redundancy Mode on the Platform Profile
2. Configure Redundancy Profile
3. Configure Session for Redundancy
4. Configure Link Aggregation Group for Redundancy

#### Configuration Syntax and Commands

The following sections describe syntax and commands for various configurations.

#### Configure the Redundancy Mode

You must select the `standby-non-forwarding` option under the platform profile.

> This setup is a one-time activity and system reboot is required the changes to take effect.

To set the desired redundancy mode, use the command.

**Syntax:**

**set system platform profile** <profile-name> **redundancy-mode standby-non-forwarding**

| Attribute | Description |
|---|---|
| `<profile-name>` | Name of the platform profile. |
| `standby-non-forwarding` | Active-Standby Redundancy mode. Standby node and LAG does not forward traffic unless a failover occurs. |

Example:

This configuration defines a platform profile named '4q' and enables 'standby-non-forwarding' mode.

```
set system platform profile 4q
set system platform profile 4q redundancy-mode standby-non-forwarding
```

```
supervisor@rtbrick.net: cfg> show config system platform profile
{
  "rtbrick-config:profile": [
    {
      "profile_name": "4q",
      "redundancy-mode": "standby-non-forwarding"
    }
  ]
}
```

#### Configuring Redundancy Profile

Redundancy profile is used to provide peer node identity in redundancy. While configuring the redundancy profile on the nodes, you need to specify IP addresses of both the peer nodes. Based on the priority value that you specify for the session ID, the peers take the roles of active and standby for the subscriber session.

**Syntax:**

**set redundancy profile** <name> **peer ipv4 instance** <instance>
**set redundancy profile** <name> **peer ipv4 remote-address** <remote-address>
**set redundancy profile** <name> **peer ipv4 update-source** <update-source>
**set redundancy profile** <name> **switchover-hold-timer** <0-600>

| Attribute | Description |
|---|---|
| `name` | Name of the redundancy profile. |
| `instance` | Name of the instance. |
| `remote-address` | Specifies the IPv4 address of the peer BNG. |
| `update-source` | Specifies the source IPv4 address used by the local node. |
| `switchover-hold-timer` | Minimum interval time between consecutive switch-overs in seconds. Supported range: 0 - 600. |

Example:

This configuration sets up a redundancy profile called 'rd_pppoe' and details the essential parameters for establishing peer connectivity. The remote-address specifies the peer BNG, the update-source designates the local address for the redundancy session, and the instance parameter defines the routing instance that manages the peer connection.

```
set redundancy profile rd_pppoe
set redundancy profile rd_pppoe peer ipv4 remote-address 198.51.100.2
set redundancy profile rd_pppoe peer ipv4 update-source 198.51.100.1
set redundancy profile rd_pppoe peer ipv4 instance default
```

```
supervisor@rtbrick.net: cfg> show config redundancy profile
{
    "rtbrick-config:profile": [
      {
        "name": "rd_pppoe",
        "peer": {
          "ipv4": {
            "remote-address": "198.51.100.2",
            "update-source": "198.51.100.1",
            "instance": "default"
          }
        }
      }
    ]
  }
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Configuring Session for Redundancy

You can configure Redundancy session with a unique session ID. You can define the system priority value (which determines active and standby roles) and the associate Redundancy profile configuration in the Redundancy Session configuration.

**Syntax:**

**set redundancy session** <id> **keepalive-interval** <value>
**set redundancy session** <id> **priority** <priority>
**set redundancy session** <id> **profile** <profile>

| Attribute | Description |
|---|---|
| `keepalive-interval` | Keepalive message transmission interval in seconds. Supported range is 1 - 30. Default is 3 seconds. |
| `id` | Session ID supported value range is 1 - 255. |
| `priority` | The node with the higher priority value becomes the active node. |
| `profile` | Redundancy profile name. |

Example for configuring session for redundancy.

The following configuration creates a redundancy session (session ID 100) and links it to the 'rd_pppoe' redundancy profile. The priority value determines the node's preference for acting as the active unit.

```
set redundancy session 100
set redundancy session 100 priority 10
set redundancy session 100 profile rd_pppoe
```

Configuration Example:

```
supervisor@rtbrick.net: cfg> show config redundancy session
{
    "rtbrick-config:session": [
      {
        "session-id": 100,
        "priority": 10,
        "profile": "rd_pppoe"
      }
    ]
  }
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Configuring LAG for Redundancy

You must associate the Redundancy session ID with the LAG. While configuring LAG for redundancy, you must specify the session ID to associate the LAG with the Redundancy Session. LAG can identify its Redundancy session with this mapping.

For information on how to configure LAG for Redundancy, see Configuring LAG for Redundancy.

#### Switchover Manually

In the Active-Standby redundancy mode, administrators can perform a manual switchover from an active node to a standby node. Use the following command to perform switchover:

```
request switch-over session <id> confirm
```

### Operational Commands

#### Redundancy Show Commands

The Redundancy show commands provide detailed information about various redundancy operations.

#### Redundancy Details

The following show commands provides information about redundancy client daemons and redundancy sessions.

**Syntax:**

**show redundancy** <options>

| Option | Description |
|---|---|
| `client` | Displays information about redundancy client daemons. |
| `session` | Displays session details. |

#### Redundancy Client Statistics

This show command displays details about the client daemon participate in the redundancy.

**Syntax:**

**show redundancy client** <name> **statistics**

This command displays information from redundancy client (daemon) which is a participant in the redundancy sessions. The daemons `ifmd` and `lagd` are the client daemons participate in the redundancy sessions.

The following output displays the statistical information for LAGD.

```
supervisor@rtbrick.net: cfg> show redundancy client lagd statistics
Session id: 100, Profile: rd_ipoe_olt1
  Agent State: down
  TCP operational state: up
  Message statistics:
  Agent State: down
  TCP operational state: up
  Message statistics:
    Keepalive sent    : 0
    Keepalive received: 0
  Last 5 state changes: [Latest first]
    down                : 2026-01-06T11:54:27.797207+0000
    demote-ready        : 2026-01-06T11:54:27.797183+0000
    demote-infra-wait   : 2026-01-06T11:54:27.797163+0000
    demote-app-wait     : 2026-01-06T11:54:27.797130+0000
    active              : 2026-01-06T10:00:02.833306+0000
  Connection statistics:
    Peer Address: 192.1.0.200
    Application down notifications: 0
    Connection down notifications: 0
    Retry count: 0
    Session down received: 3
Session id: 200, Profile: rd_ipoe_olt2
  Agent State: down
  TCP operational state: up
  Message statistics:
    Keepalive sent    : 0
    Keepalive received: 0
  Last 5 state changes: [Latest first]
    down                : 2026-01-06T11:54:31.831395+0000
    demote-ready        : 2026-01-06T11:54:31.831373+0000
    demote-infra-wait   : 2026-01-06T11:54:31.831354+0000
    demote-app-wait     : 2026-01-06T11:54:31.831324+0000
    stand-alone         : 2026-01-06T11:54:27.988625+0000
  Connection statistics:
    Peer Address: 192.1.0.20
    Application down notifications: 0
    Connection down notifications: 0
    Retry count: 0
    Session down received: 3
```

#### Redundancy Session Details

These show commands provide information about redundancy sessions.

**Syntax:**

**show redundancy session** <id> <options>

| Option | Description |
|---|---|
| - | Without any option, it displays all session details. |
| detail | Displays details session information. |
| history | Displays session history. |
| status | Displays session status. |

Example output showing the status of a specified redundancy session ID.

```
supervisor@rtbrick.net: cfg> show redundancy session 100 status
State: down,  Remote State: down
Redundancy client replication information:
  Total redundancy clients : 4
  ifmd:
  ipoed.1:
  lagd:
  subscriberd.1:
    Number of subscribed table: 2
    Number of published table: 0
    Number of table subscription requested: 2
```

The following example output displays the detailed view of the specified redundancy session.

```
supervisor@rtbrick.net: cfg> show redundancy session 200 detail
Redundancy session ID: 200, Source: 192.1.0.10, Peer: 192.1.0.20
  Instance: default, Profile name: rd_ipoe_olt2, Local priority: 210
  State: down, Previous state: stand-alone, Last state transition time: 2026-01-
06T11:54:31.825860+0000
  Peer state: down
  Mode: standby-non-forwarding
  Interface: lag-11
  TCP operational state: up
  Message statistics:
    Keepalive sent       : 13248
    Keepalive received   : 13249
    Keepalive ignored    : 0
  Switchovers via cli    : 0
  State transition statistics:
    Down                 : 1
    Connect              : 1
    Standby              : 1
    Active               : 0
    Stand-alone          : 1
    Demote               : 0
```

```
  Timers:
    Connect retry interval              : 4294967295
    Keepalive timer interval            : 5000
    Holddown timer interval             : 20000
    Switchover timer pending interval   : 0 sec
  Timer Statistics:
    Holddown timer expiry count         : 0
```

The following example output displays the recent history entries for redundancy session 200, limited to 5 events

```
supervisor@rtbrick.net: cfg> show redundancy session 200 history count 5
Previous state       Current state        State change reason  Timestamp
stand-alone          down                 down                 2026-01-
06T11:54:31.825860+0000
standby              stand-alone          stand-alone          2026-01-
06T11:54:27.983076+0000
connect              standby              standby              2026-01-
06T10:00:02.627726+0000
down                 connect              open                 2026-01-
06T09:59:52.767142+0000
invalid              down                 init                 2026-01-
06T09:59:06.689775+0000
```

# 6. Forwarding

The Forwarding section of the RtBrick FullStack (RBFS) User Guides details the configuration and operational commands for visibility and management of data plane traffic.

Key Features:

- Includes documentation for configuring basic network Interfaces and handling neighbor discovery via ARP/ND.
- Details how to apply network security parameters using Access Control Lists (ACLs) and inspect traffic patterns using Port Mirroring.
- Provides instructions for bundling multiple physical network links into a single logical channel to increase bandwidth and offer redundancy.
- Explains how to optimize data plane traffic behavior using Hierarchical Quality of Service structures to enforce granular traffic scheduling and shaping.
- Covers neighbor discovery and hardware tracking capabilities through the Link Layer Discovery Protocol (LLDP).

The following guides available under the Forwarding section:

- Interfaces
- ARP/ND
- ACLs
- Port Mirroring
- LAG
- HQoS
- LLDP

## 6.1. Interfaces

### 6.1.1. Interfaces Overview

RtBrick Full Stack (RBFS) supports various types of interfaces, including physical and logical interfaces. On hardware platforms, RBFS physical interfaces represent the ports of a switch. This guide describes how to configure and verify RBFS interfaces. Features like routing protocols or access services will typically run on top of the interfaces.

#### Interface Types

**Physical Interfaces**

In RBFS, physical interfaces (IFP) typically represent the physical ports of a hardware switch. For example, ifp-0/0/1 represents switch port 1. On the physical interface level, you can configure various parameters associated with Layer 1 of the ISO/OSI reference model.

**Logical Interfaces**

For each physical interface, you can create one or multiple interface units also referred to as logical interfaces (IFL) in RBFS. A logical interface is associated with the Layer 2 operation. In addition, you can configure Layer 3 parameters like IP addresses on interface units, and assign interface units to routing instances.

**Loopback Interfaces**

A loopback interface is typically used to represent and identify a device itself. Loopback interfaces are preferred because they do not depend on the status of a physical port, and will always be up. Please note, although loopback interfaces are virtual interfaces, there are also represented as physical interfaces and interface units in RBFS, reflecting Layer 1 and Layer 2/3 operation. Do not configure the broadcast IP address on the loopback interface.

**Host Interfaces**

Linux virtual ethernet (veth) interfaces connect an LXC container with the Linux host OS. In RBFS, a veth interface to the Linux bridge lxcbr0 is created by default. In virtual topologies, you can create additional veth interfaces and Linux bridges. RBFS host interfaces represent veth interfaces in RBFS.

For example, if the container interface eth1 connects to the host interface vethXYZ123, ifp-0/0/1 can be bound to eth1 to represent it in RBFS. Host interfaces can be used like any other physical interface.

**Memory Interfaces**

Memory interfaces (memif) are virtual interfaces used for creating virtual topologies. They connect multiple containers running RBFS to each other. When configuring memif interfaces:

- Endpoints match on the memif ID, i.e. the memif ID needs to be the same on both ends.
- memif IDs need to be unique on the host.
- The memif interface name is locally significant only.
- One endpoint needs to be configured as a master, while the other one is configured as a slave.

**Interface Numbering**

RBFS interface numbers match the port numbers on the switch faceplate. An interface is named in the ifp-<chassis-ID>/<front-panel-block-number>/<port> format. For example, ifp-0/0/1.

- Chassis ID—always 0 for the currently supported platforms
- Front Panel Block—represents group of ports on the faceplate
- Port—matches the port number on switch faceplate

Virtual interfaces follow the same structure, for example, `lo-0/0/1` or `memif-0/0/1`.

Logical interfaces are numbered: `ifl-<Node ID>/<Chip ID>/<Port ID>/<Unit ID>`, for example `ifl-0/0/1/1`.

### Logical Interface Unit Limit

The supported logical interface (IFL) unit number is 16,000. It provides flexibility in naming logical interfaces across a variety of service configurations and is useful for deployments that require large numbers of logical interfaces, such as double-tagged VLAN environments that combine Layer 2 and Layer 3 sub-interfaces.

### Reverse Path Forwarding on Logical Layer-3 Interfaces

RBFS supports enabling Reverse Path Forwarding (RPF) in both strict and loose modes on logical Layer-3 IFLs. This feature ensures that packets received on IFLs can be validated against their expected reverse path. It improves security for deployments that rely on Layer 3 IFLs.

This feature is supported on all Q2C and Q2A-based RBFS platforms.

### Community Support for Interfaces

You can tag an interface address with a community or extended community. RBFS will create a direct route for each interface address. If a community or extended community is configured for an interface address, RBFS will add it to the direct route. Communities can be used in policies. For example, when redistributing direct routes, you can match these communities and define desired policy rules.

### Unnumbered Interfaces

An unnumbered interface is a point-to-point interface that is not explicitly configured with a dedicated IP address; instead, it borrows an IP address from a loopback interface.

For details about configuring an unnumbered interface, see section Logical Interface Configuration.

#### Unnumbered Interface to Numbered Interface (and Vice versa) Conversion

RBFS supports the conversion of unnumbered to numbered interface and vice versa in a single commit. The existing address should be deleted before converting the interface to an unnumbered interface. These two operations are supported in one commit.

#### Converting a numbered interface to an unnumbered interface

```
supervisor@rtbrick>: cfg> delete interface ifl-0/0/0 unit 11 address ipv4 192.0.2.1/24
supervisor@rtbrick>: cfg> set interface ifl-0/0/0 unit 10 unnumbered interface lo-0/0/0/0
supervisor@rtbrick>: cfg> commit
```

#### Converting an unnumbered interface to a numbered interface

```
supervisor@rtbrick>: cfg> del interface ifl-0/0/3 unit 10 unnumbered interface lo-0/0/0/0
supervisor@rtbrick>: cfg> set interface ifl-0/0/3 unit 10 address ipv4 192.0.2.1/24
supervisor@rtbrick>: cfg> commit
```

#### Interface Description Length

Interface descriptions support up to 1024 characters (previously 128). This applies to all regular IFP, Layer 3 IFL, Layer 2 IFL, LAG Layer 3 IFL, and LAG Layer 2 IFL interfaces.

### Interface States

RBFS supports various interfaces such as physical interface, logical interface, and LAG interface and it uses various indicators to show the various states of interfaces. All interfaces have different states such as the following:

- Admin state: Indicates whether the interface is enabled (Up) or disabled (Down).
- Link state: Indicates whether the interface is linked (Up) or not linked (Down).
- Oper state: Indicates whether the interface is functional (operational) or not functional.
- IPv4 state: Indicates that the interface is configured with the IPv4 address.
- IPv6 state: Indicates that the interface is configured with the IPv6 address.
- MPLS state: Indicates that the interface is an MPLS-enabled interface.

#### Physical Interface States:

The physical interface states include:

- admin
- oper (operational)
- link

#### Logical Interface States:

The logical interface states include:

- admin
- oper (operational)
- link
- IPv4
- IPv6
- MPLS

### LAG Interface States

The LAG interface states include:

- admin
- oper(operational)

The interface state can either be Up or Down. Up shows that it is ready to pass packets, and Down shows that it is not ready to transmit packets. The operational state shows that the interface is

operational and is ready to transmit packets. The `Oper` state is Up only if both the Admin state and Link state are Up.

### Auto-negotiation

Auto-negotiation allows directly connected devices to automatically exchange speed and duplex mode information for the links. If auto-negotiation is enabled, ports can auto-negotiate the speed and duplex capabilities with other ports. The auto-negotiation can determine the best speed and duplex at which the ports can operate optimally.

> Port speed configuration and auto-negotiation are mutually exclusive.

RBFS supports auto-negotiation between ports in the following ways:

- 1G ports can negotiate with 10G ports.
- 40G ports can negotiate with 100G ports.

Auto-negotiation is not supported for the following combinations:

- 40G ports cannot negotiate with 1G ports, 10G ports, and 25G ports.
- 100G ports cannot negotiate with 1G ports, 10G ports and 25G ports.
- 1G port cannot negotiate with 25G port.

### Interface Counters

Interface counters are statistics that network devices maintain for the traffic passing through the interfaces for all physical, logical, and LAG interfaces on a device. These counters provide information about the utilization and performance of the interface. Users can identify and troubleshoot issues such as congestion, errors, and performance bottlenecks by monitoring these counters.

#### RBFS Logical Interface counters and descriptions

| Counters | Descriptions |
|---|---|
| Rx packets | The number of IP packets received by the logical interface. |
| Rx bytes | The number of bytes received by the logical interface. |
| Tx packets | The number of packets transmitted by the logical interface. |
| Tx bytes | The number of bytes transmitted by a logical interface. |
| IPv4 packets | The number of IPv4 packets processed by a logical interface. |
| IPv6 packets | The number of IPv6 packets processed by a logical interface. |
| MPLS packets | The number of MPLS packets processed by a logical interface. |
| Punt packets | The number of packets that are punted or forwarded to the CPU for further processing by a logical interface. |
| Drops packets | The number of packets that were dropped by a logical interface. |

| Counters | Descriptions |
|---|---|
| Rx Miss packets | The number of packets that were dropped by a logical interface. |
| Rx Error packets | The number of packets that were received with errors by the logical interface. |
| Rx No Buff packets | The number of packets that could not be received due to a lack of buffer space. When a logical interface receives a packet but does not have enough buffer space to store it, the packet is dropped. |
| Tx Error packets | The number of packets that could not be transmitted due to errors encountered during the transmission process. |
| **Packet Statistics** | |
| Ingress forwarded packets | The number of packets that are received by the device on one interface and then forwarded out on another interface. |
| Ingress forwarded bytes | The number of bytes that are received by a device on one interface and then forwarded out on another interface. |
| Ingress drop Packets | The number of packets that are dropped (discarded) by a network device upon arrival on one of its interfaces by a device. |
| Ingress drop bytes | Total number of bytes that have been dropped by an interface on incoming traffic. |
| Egress forwarded packets | Total number of packets that have been successfully forwarded by an interface on outgoing traffic. |
| Egress forwarded bytes | The number of bytes that have been forwarded by an interface in the egress direction. |
| Egress drop packets | The number of packets that have been dropped by an interface in the egress (outgoing) direction. |
| Egress drop bytes | The number of bytes that have been dropped by an interface in the egress direction. |

**RBFS physical interface counters and descriptions**

| Physical Interface Counters | Descriptions |
|---|---|
| **VPP Statistics** | |
| Rx packets | The number of IP packets received by the physical interface. |
| Rx bytes | The number of bytes received by the physical interface. |
| Tx packets | The number of packets transmitted by the physical interface. |
| Tx bytes | The number of bytes transmitted by the physical interface. |
| IPv4 packets | The number of IPv4 packets processed by the physical interface. |
| IPv6 packets | The number of IPv6 packets processed by the physical interface. |
| MPLS packets | The number of MPLS packets processed by a physical interface. |

| Physical Interface Counters | Descriptions |
|---|---|
| Punt packets | The number of packets that are punted or forwarded to the CPU for further processing by a physical interface. |
| Drops packets | The number of packets that were dropped by a physical interface. |
| Rx Miss packets | The number of packets that were dropped by a physical interface. |
| Rx Error packets | The number of packets that were received with errors by the physical interface. |
| Rx No Buff packets | The number of packets that could not be received due to a lack of buffer space. When a physical interface receives a packet but does not have enough buffer space to store it, the packet is dropped. |
| Tx Error packets | The number of packets that were dropped by a physical interface. |
| **BCM Statistics** | |
| inOctets | Number of octets received through the interface. |
| inUcastPkts | Number of unicast packets received through the interface. |
| inNonUcastPkts | Number of non-unicast packets received through the interface. |
| inErrors | The number of inbound packets that contained errors preventing them from being processed correctly. |
| outOctets | Number of octets sent through the interface. |
| ifHCInOctets | The number of octets (8-bit bytes) received by a network interface. It is a part of the SNMP MIB (Management Information Base) structure and is used to track high-capacity input octets counter used in the context of SNMP monitoring. |
| outUcastPkts | Number of unicast packets sent through the interface. |
| outNonUcastPkts | Number of non-unicast packets sent through the interface. |
| outErrors | The number of outbound packets that contained errors preventing them from being processed correctly. |
| etherStatsDropEvents | Number of events where packets were not delivered to the protocol stack because of resource limitations or other reasons. These dropped events can occur due to buffer overflows, congestion, or hardware limitations. |
| etherStatsMulticastPkts | The number of packets received by an interface that were addressed to a multicast MAC address. |
| etherStatsBroadcastPkts | The number of broadcast packets received on an Ethernet interface. |

| Physical Interface Counters | Descriptions |
| --- | --- |
| etherStatsUndersizePkts | The number of received packets that are smaller than the minimum allowed Ethernet frame size. Undersized packets can indicate various issues. |
| etherStatsFragments | The number of received packets that are fragments of IP datagrams. |
| etherStatsOversizePkts | The number of received packets that exceed the maximum Ethernet frame size. |
| etherStatsOctets | The total number of octets (bytes) of data transmitted and received on an Ethernet interface. This counter provides a measure of the total amount of data traffic on the interface. |
| etherStatsPkts | The number of packets transmitted or received by an Ethernet interface. This counter can provide insights into the traffic load and performance of the interface. |
| dot1dBasePortMtuExceededDiscards | The number of frames that were discarded at an interface as they exceeded the Maximum Transmission Unit (MTU) of the port. This typically happens when a frame is larger than the maximum size allowed on the interface and cannot be fragmented, so it is dropped. |
| etherStatsTXNoErrors | The number of Ethernet frames transmitted without any errors through the Ethernet interface. Each time a frame is successfully transmitted without encountering any errors, this counter is incremented. |
| etherStatsRXNoErrors | The number of Ethernet frames received without any errors through the Ethernet interface. Each time a frame is successfully transmitted without encountering any errors, this counter is incremented. |
| inMulticastPkts | The number of packets received by the interface that were addressed to a multicast address. These counters provide insights into the amount of multicast traffic being received by the interface. |
| outBroadcastPkts | The number of packets transmitted by the network interface as broadcast packets. These counters can provide insights into the amount of broadcast traffic generated by the interface. |
| outMulticastPkts | The number of packets transmitted by the interface as multicast packets. These counters can provide insights into the amount of multicast traffic generated by the interface. |
| outBroadcastPkts | The number of packets transmitted by the interface as broadcast packets. These counters can provide insights into the amount of broadcast traffic generated by the interface, which can be useful for network troubleshooting and monitoring network performance. |

| Physical Interface Counters | Descriptions |
| --- | --- |
| bcmReceivedUndersizePkts | The number of undersized packets received by a Broadcom device. Undersized packets are Ethernet frames that are smaller than the minimum allowed size. This counter provides insights into packet size issues. |
| bcmTransmittedUndersizePkts | The number of undersized packets sent by a Broadcom device. Undersized packets are Ethernet frames that are smaller than the minimum allowed size. This counter provides insights into packet size issues. |
| etherTxOversizePkts | The number of packets that exceed the maximum transmission unit (MTU) size allowed on an Ethernet interface. |
| etherStatsJabbers | The number of jabber frames received by an Ethernet interface. Jabber frames are Ethernet frames that exceed the maximum allowed frame size and contain data that extends beyond the maximum length. |
| etherStatsCRCAlignErrors | The number of frames received by an Ethernet interface that has a CRC (Cyclic Redundancy Check) error. Also, the frames are not an integral number of octets in length; that is, the frame length is not a multiple of 8 bits.<br><br>CRC errors occur when the CRC checksum calculated by the receiving interface does not match the CRC checksum transmitted by the sending interface, indicating that the frame may have been corrupted during transmission |
| dot3StatsFCSErrors | The number of frames that have a Frame Check Sequence (FCS) error received by an Ethernet interface. The FCS is a field in the Ethernet frame that contains a checksum calculated based on the contents of the frame. |
| ifHCOutMulticastPkts | The number of outbound multicast packets on a network interface. The ifHCOutMulticastPkts object uses a 64-bit counter, allowing it to accommodate high-speed interfaces without wrapping around as quickly. It provides an accurate count of the outbound multicast packets on the interface. |
| ifHCOutBroadcastPckts | The ifHCOutBroadcastPkts counter provides the number of outbound broadcast packets on an interface. It is part of the IF-MIB (Interface MIB) and is an extension of the standard ifOutBroadcastPkts counter, providing a 64-bit counter for high-speed interfaces to avoid wrapping around quickly. |

**Permanent ARP Entry on IFLs**

The Permanent ARP Entry functionality is supported on IFLs. This functionality enables devices to proactively resolve ARP and ND.

**Path MTU Discovery**

Path MTU Discovery (PMTUD) is a technique that is used to determine the Maximum Transmission Unit (MTU) size that can be used along the path from a source to a destination without requiring fragmentation. PMTUD helps to avoid fragmentation by dynamically discovering the smallest MTU in the path.

PMTUD is crucial for optimizing network performance by avoiding fragmentation.

By default, path MTU discovery is enabled in RBFS.

When RBFS receives packets that violate the MTU, meaning their size exceeds the allowed MTU size, it will respond with the following ICMP error message:

**ICMPv4 Message Types**

The type field specifies the message type sent by the host, providing detailed information about the error condition.

The following tables show the ICMPv4 message types.

| Type | Description |
|---|---|
| 3 | Destination Unreachable.<br>This message notifies the source host about delivery issues faced while attempting to reach the destination. |

Destination Unreachable uses the following code values to further describe the function of the ICMP message being sent.

| Code | Description |
|---|---|
| 4 | Fragmentation Needed and Don’t-Fragment (DF) was Set.<br>This message shows when a router receives a packet that requires fragmentation while having the DF flag enabled. |

**ICMPv6 Message Types**

The type field identifies the type of the message sent by the host. The type field contains more specific information about the error condition.

The table below lists the ICMPv6 message type.

| Type | Description |
|---|---|
| 2 | Packet Too Big.<br>A Packet Too Big must be sent by a router when it receives a packet that cannot be forwarded because it is larger than the MTU of the outgoing link. |

| Code | Description |
|---|---|
| 0 | No code |

You can modify this behavior by enabling fragmentation. For details on how to enable hostpath fragmentation, refer to the section 'Enabling Hostpath Fragmentation' below.

All outgoing packets are validated against the configured MTU on the egress path.

- If MTU is violated and MTU-profile action is `drop`, then packets are dropped in hardware.
- If MTU is violated and MTU-profile action is `redirect-to-cpu`, a 20MB policer is used to protect the CPU-port from overwhelming MTU-violated traffic, and packets are sent to the CPU port.
- When fragmentation is enabled, one of the following operations takes place.
  - If the DF (Don't Fragment) bit is not set in the received packet (applicable only to IPv4), the packets are fragmented and sent to the outgoing port.
  - If the DF bit is set in a packet, it drops the packet and sends an ICMP error message back to the source host.
- If the fragmentation is disabled, packets are dropped and ICMP error messages are sent to the source host.

For information on configuring the MTU profile, see MTU Profile Configuration.

### IP Fragmentation

If the size of the original IP packet exceeds the MTU of the outgoing interface, the packet must be fragmented.

> RBFS supports IP fragmentation on its Q2C platforms.

The MTU profile configuration determines if packets exceeding the MTU are dropped or sent to the CPU for further processing.

The `forwarding-options` configuration includes a setting to enable hostpath fragmentation for packets redirected to the CPU. If this setting is not configured, the device responds with an ICMP error message to the source host.

> MTU fragmentation is not supported on core interfaces configured as LAG. You must disable the re-injection for MTU fragmentation to function on LAG interfaces nn Q2C platforms.

#### Guidelines and Limitations of IP Fragmentation

The following guidelines and limitations apply to IP Fragmentation:

- If a packet exceeds the negotiated subscriber MTU size, the packet is either dropped or fragmented based on the configured MTU profile (applicable only to the Q2C platforms). Both fragmentation and ICMP error messages are supported only for subscriber (PPPoE & IPOE) downstream traffic. For details, see section MTU Profile Configuration.

You can control fragmentation behavior using the `set forwarding-options fragmentation ipv4 state CPU` command. For more information about configuring fragmentation, see Enabling Hostpath Fragmentation. * Fragmented packets are reinjected into the egress pipeline and forwarded

through the regular QoS path.

- No ICMP error messages are generated for MTU-violated multicast packets.

**MTU Profile**

The Maximum Transmission Unit (MTU) defines the largest packet size that is allowed to transmit across the network.

In the Q2C Platforms, resource efficiency is achieved by using MTU profiles. Components such as IFPs, IFLs, and L3 interfaces reference these MTU profiles. To optimize MTU resource management, you can configure MTU profiles, which can then be applied to various attachment points. The MTU profiles are attached to the physical (IFP), logical (IFL) and L3 interfaces. RBFS supports the following attachment points for MTU profiles.

**Attachment Points:** The MTU profiles are attached to the interface entities such as physical (IFP), logical (IFL) and L3 interfaces. RBFS supports the following attachment points for MTU profiles:

- Port-level
- L3 interface level (IPv4 and IPv6)
- PPPoE subscriber level (L2 IFL)
- IPoE subscriber level

MTU Size: A user-configured MTU size can range from 64 to 9216 in RBFS.

> For MTU profiles of type 'pppoe', users should provide L3 MTU size (IPv4/IPv6 headers).

MTU Type: An MTU type specifies the attachment point of the MTU profile. The MTU types supported are as follows:

- **physical**: When checking the MTU, the total size of the packet, including headers, is considered.
- **ipv4**: The MTU check is conducted based on IPv4 headers.
- **ipv6**: The MTU check is conducted based on IPv6 headers.
- **ip**: MTU profile of type IP and IPoE.
- **pppoe**: The MTU profile is applied to the PPPoE subscriber interface, where you must specify the L3 MTU size. Using its best-match algorithm, the subscriber management service associates these profiles with PPPoE subscribers.

> - MTU profiles for Layer 3 logical interfaces must be explicitly configured. If not set, no default MTU size is applied for either IPv4 or IPv6 traffic.
> - On the Q2C platform, L3 interfaces can be configured with either an IPv4 MTU profile or an IPv6 MTU profile, but not both. However, by using an MTU profile of type 'ip', you can set a common MTU size that applies to both IPv4 and IPv6 traffic.
> - On the QAX platform, only physical MTU profiles are supported.

**MTU Action:**

The MTU action specifies the response when an MTU check fails. The following actions are supported:

- `drop`: This indicates that when the MTU check fails, the action "drop" is performed.
- `redirect-to-cpu`: This is an action of redirecting packets to the CPU in a traffic behavior. The `redirect-to-cpu action` must be configured for the fragmentation to work.

#### MTU Profile Limitations

The MTU profile has the following limitations.

- The MTU violation trap is signalled at the ETTP stage (Egress) of the pipeline. Therefore, it is expected that the bandwidth consumed by dropped packets due to MTU violations will be accounted for.
- Each hardware unit supports a limited number of MTUs. Therefore, the following limits apply.
- **MTU Profile Limits by Platform:**
  - **Q2C Platform:**
    - Total MTU profiles supported: 8
    - L3 MTU profiles (IP/IPv4/IPv6): Up to 3
    - PPPoE MTU profiles: Up to 6 (including the default PPPoE profile)
    - Physical MTU profiles: Up to 7
  - **QAX Platform:**
    - Physical MTU profiles: 1 only

### Interface Holddown

#### Overview

Interface Holddown or Interface Dampening is a delay mechanism to prevent network instability from flapping interfaces. Interface flapping occurs when an interface goes up and down repeatedly, leading to excessive route recalculations and needless notifications to other software components.

The Interface Holddown feature is used to prevent network instability by temporarily damping the interface state if it is fluctuating. This mechanism suppresses frequently fluctuating routes and ensures that only stable routes are advertised. It improves overall network performance by effectively utilizing resources such as memory and processing power.

> The Interface Holddown feature remains supported when interfaces are grouped in a LAG.

### Option to Disable

By default, the Interface Holddown feature is enabled on RBFS. You can configure the ‘hold-down down delay’ and ‘hold-down up delay’ parameter values using the configuration commands. You

can also disable the feature using the command.

#### Benefits of Interface Holddown

Frequent updates due to route flaps can consume significant processing capacity. This can increase the CPU load on routers, and the constant route advertisements and withdrawals can consume bandwidth. The Interface Holddown mechanism minimizes unnecessary route recalculations and notifications, and the number of updates that need to be processed by suppressing frequently flapping routes. Suppressing unstable routes can improve overall network performance by ensuring that only stable routes are advertised and used for forwarding traffic.

#### How RBFS Interface Holddown Works

When an interface flaps, such as an interface goes up and down frequently, RBFS employs a holddown period during which the interface's status is retained in its previous state.

The holddown time for an interface increases exponentially within a defined range. The values can be configured explicitly for each interface. When an interface flap occurs, the interface management daemon receives this event from the hardware through the FIBD.

The following diagram shows the interface holddown feature.

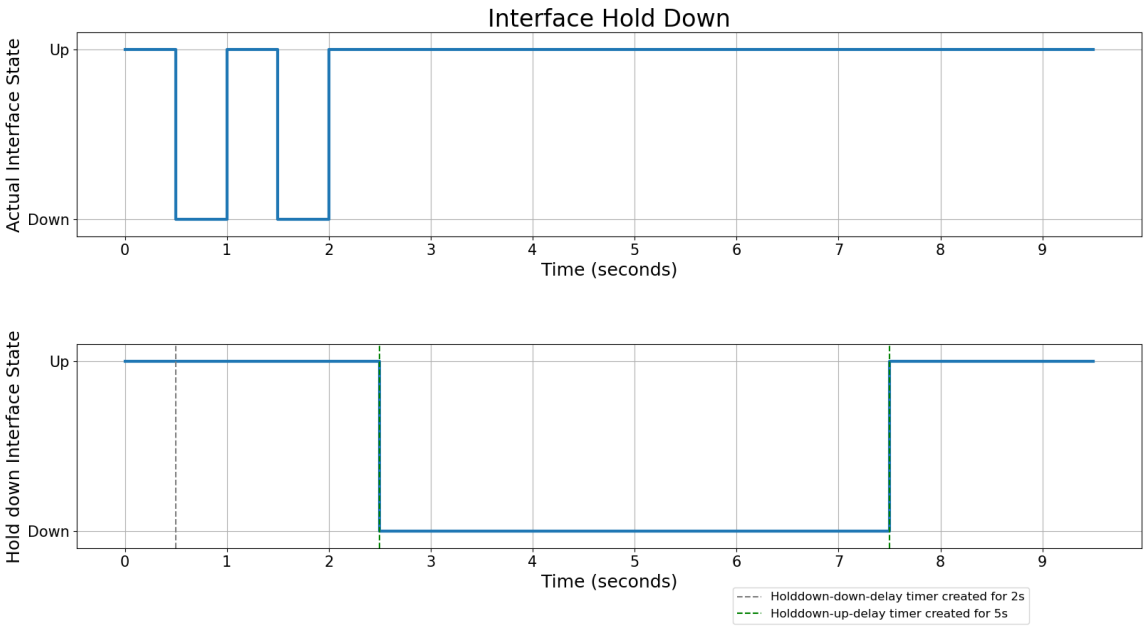



*Figure 110. Interface Holddown*

The holddown intervals are set at 0ms, 100ms, 200ms, 500ms, 1s, 2s, 5s, and 10s. When the holddown timer expires, the interface state is propagated to higher layers. If the interface remains stable throughout the holddown interval, the actual state is propagated. However, if the interface experiences instability during this interval, it remains in the DOWN state.

RBFS implements "Holddown Down Delay" and "Holddown Up Delay", two parameters that control how the interface flap is managed.

### Holddown Down Delay

The ‘Hold-down Down Delay’ parameter, once configured, ensures that the interface state change from UP to DOWN is delayed by the specified ‘holddown down delay’ period. After this delay, the default holddown mechanism takes over. Specifically:

1. The ‘holddown’ interval starts with its initial value, that is 'holddown down delay' period, and increments exponentially when each timer expires.
2. DOWN to UP state changes are delayed based on exponentially increasing intervals.

The following diagram shows 'holddown down delay' scenario.

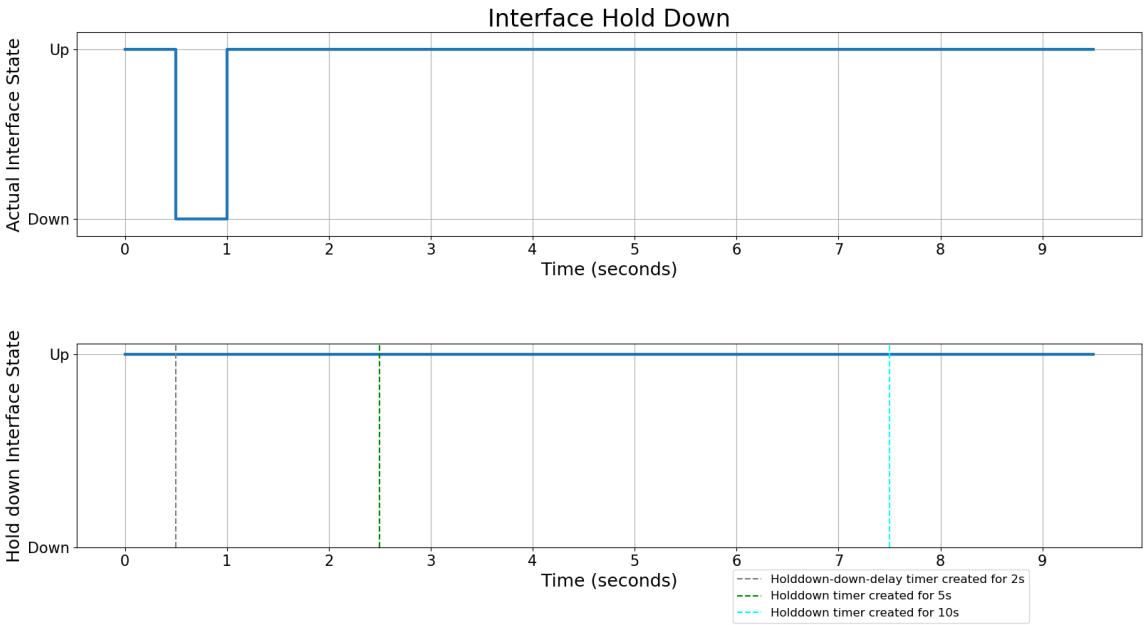



*Figure 111. Interface Holddown Delay*

For example, if the ‘holddown down delay’ is set to 5 seconds and the interface state changes from UP to DOWN, the interface remains up for 5 seconds. After this 5-second delay, the system checks the stability of the interface:

- If the interface has no flap during this period, it remains up.
- If the interface has flapped during this period, it changes to the DOWN state after the delay.

In summary, Holddown Down Delay ensures an interface remains up for a specified time before propagating any state change to higher-level software components.

**Holddown Up Delay**

'Holddown Up Delay' is the period that an interface stays down after a failure before it is considered to be in a stable state. When the state changes from DOWN to UP, a holddown timer is initiated with a specified delay period. If there are flaps during the previous interval time, then a new timer with the same 'holddown up' delay is set.

This results in a delay in the state transitioning to UP based on the ‘holddown up delay’. However, if state changes happen within the 'holddown up delay' interval, any change from UP to DOWN is immediately transmitted. It uses a static timer instead of an exponential backoff for the holddown

interval.

NOTE:

> If the holddown-up delay of 10,000 ms is explicitly configured by the user, the system will delay the link's status change by a fixed value of 10,000 ms. For example, if the link goes down and later becomes up, the system will declare the link as 'up' only after the delay of 10,000 ms.
>
> If the holddown-up delay of 10,000 ms is not explicitly configured by the user, the default setting will show a value of 10,000 ms in the interface details. This default value represents the maximum duration that the system checks for link stability before declaring it as 'up'. In this case, if the link goes down and later returns, the system may declare the link as up immediately if the link is stable. However, if the link is not stable, the holddown time for the interface will increase exponentially within the range of 10,000 ms before it finally declares the link as 'up'.

The following diagram shows 'holddown up' scenario.

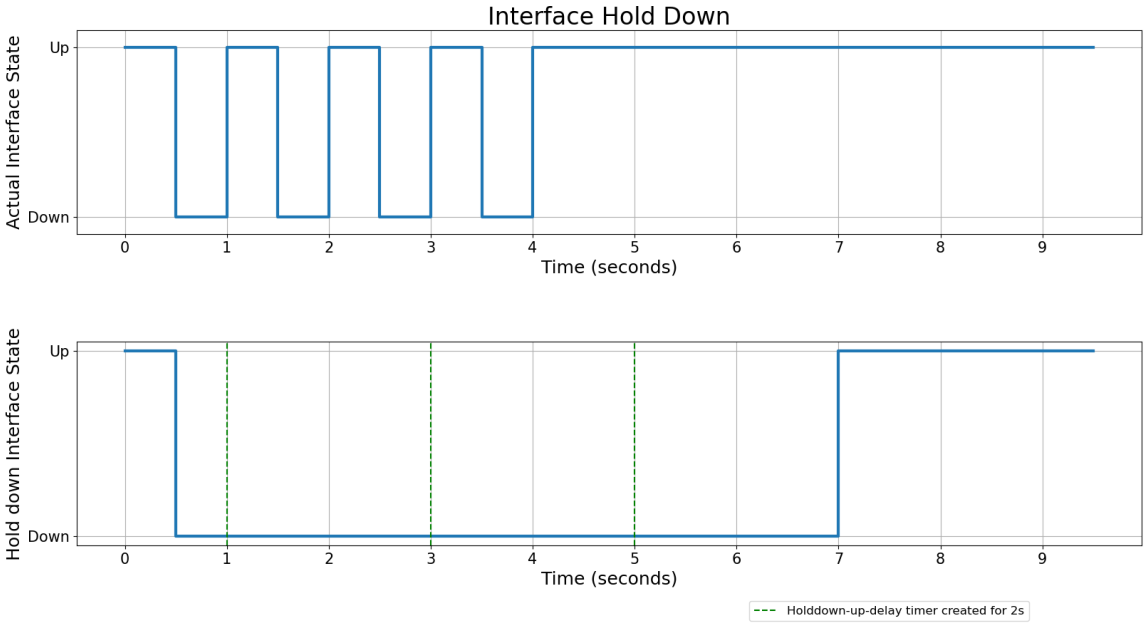



*Figure 112. Interface Holddown Up Scenario*

**Example Scenario**

Imagine where a router's interface occasionally loses and regains connectivity due to a loose cable. Each time this happens, it creates an event that propagates through many of the software components in the network.

Without the Interface Holddown feature, every time disruption immediately triggers network protocols and higher-level software components to respond, potentially causing routing updates, recalculations, and additional processing overhead. This constant back and forth leads to network instability and adds additional strain on network devices.

With the interface Holddown enabled, the RBFS still receives these flap events but employs a hold-down mechanism to delay taking action on these changes. For example, with the holddown intervals: 0ms, 100ms, 200ms, 500ms, 1s, 2s, 5s, and 10s, the system performs the following:

- Retain the old interface status when an interface flap is detected.
- Wait for the initial hold-down time, which is 100ms.
- If another transition occurs within this period, the hold-down time increases to the next interval that is, 200ms. This process repeats, increasing the hold-down time after each transition up to the configured maximum value, which is 10s.

In this way, by delaying the processing of interface flaps, interface dampening reduces the impact of flapping on the network. The system only updates the interface status after a stable period without transitions, thus preventing frequent, unnecessary updates.

#### IPv6 Link-Local Address

In IPv6 networking, link-local addresses are essential for local communication between nodes on the same link. It facilitates communication between nodes on the same link, however cannot be used to send packets beyond that link.

In RBFS, a link-local IPv6 address can be either automatically generated or manually assigned.

When a configured link-local address is deleted, a unique link-local address is automatically generated for the logical interface. This automatically assigned address is typically derived from the interface's MAC address and other parameters.

You can also modify an existing automatically generated link-local address by manually configuring a link-local IPv6 address on a logical interface. It will replace the auto-generated link-local address with the one you have configured.

### Dedicated Policer for IPv6 Link-Local

RBFS supports a dedicated 50 Mbps policer that handles IPv6 link-local and control traffic, including link-local (FE80), all-nodes (FF01), and all-routers (FF02) multicast traffic used by Router Advertisement (RA) and Neighbor Solicitation (NS). It ensures reliable policing of essential IPv6 control traffic.

## Guidelines & Limitations

### QAX-based Platforms

An additional restriction applies to ports on QAX-based platforms: because of hardware design, physical ports are grouped into quads (groups of 4, also known as port groups). Each quad must have the same physical parameters: speed, link-training, duplex.

The following tables are provided for easy identification of ports that need to have the same physical settings:

## Edgecore 7316-26XB Port Groups:

| Port | Speed | Duplex | Port Group |
|---|---|---|---|
| ifp-0/0/0 | 100G | Full | 0 |
| ifp-0/0/1 | 100G | Full | 1 |
| ifp-0/1/0 | 10G | Full | 2 |
| ifp-0/1/1 | 10G | Full | |
| ifp-0/1/2 | 10G | Full | |
| ifp-0/1/3 | 10G | Full | |
| ifp-0/1/4 | 10G | Full | 3 |
| ifp-0/1/5 | 10G | Full | |
| ifp-0/1/6 | 10G | Full | |
| ifp-0/1/7 | 10G | Full | |
| ifp-0/1/8 | 10G | Full | 4 |
| ifp-0/1/9 | 10G | Full | |
| ifp-0/1/10 | 10G | Full | |
| ifp-0/1/11 | 10G | Full | |
| ifp-0/1/12 | 10G | Full | 5 |
| ifp-0/1/13 | 10G | Full | |
| ifp-0/1/14 | 10G | Full | |
| ifp-0/1/15 | 10G | Full | |
| ifp-0/1/16 | 25G | Full | 6 |
| ifp-0/1/17 | 25G | Full | |
| ifp-0/1/18 | 25G | Full | |
| ifp-0/1/19 | 25G | Full | |
| ifp-0/1/20 | 25G | Full | 7 |
| ifp-0/1/21 | 25G | Full | |
| ifp-0/1/22 | 25G | Full | |
| ifp-0/1/23 | 25G | Full | |

## UfiSpace S9500-22XST Port Groups:

| Port | Speed | Duplex | Port Group |
|---|---|---|---|
| ifp-0/0/0 | 10G | Full | 8 |
| ifp-0/0/1 | 10G | Full | |
| ifp-0/0/2 | 10G | Full | |
| ifp-0/0/3 | 10G | Full | |
| ifp-0/0/4 | 10G | Full | 4 |
| ifp-0/0/5 | 10G | Full | |
| ifp-0/0/6 | 10G | Full | |
| ifp-0/0/7 | 10G | Full | |
| ifp-0/0/8 | 10G | Full | 11 |
| ifp-0/0/9 | 10G | Full | |
| ifp-0/0/10 | 10G | Full | |
| ifp-0/0/11 | 10G | Full | |
| ifp-0/0/12 | 25G | Full | 3 |
| ifp-0/0/13 | 25G | Full | |
| ifp-0/0/14 | 25G | Full | |
| ifp-0/0/15 | 25G | Full | |
| ifp-0/0/16 | 25G | Full | 2 |
| ifp-0/0/17 | 25G | Full | |
| ifp-0/0/18 | 25G | Full | |
| ifp-0/0/19 | 25G | Full | |
| ifp-0/0/20 | 100G | Full | 0 |
| ifp-0/0/21 | 100G | Full | 1 |

A PHY Quad can be associated with network interface (NIF) ports of identical type only. For example, a quad cannot be a mix of XLGE and XE ports. An exception is GE and XE ports which can coexist in the same quad. This means all the ports in a port group should have the same physical interface configuration (that is, speed/duplex/link-training). Ports in a port group are only allowed to support 1G and 10G speeds; any other combination is not allowed. If a port within a port group is misconfigured, then it would require changing the speeds/interface type of all ports within the port group to a different type and then back into the original type.

#### Support for TTL Inheritance from IP to MPLS

RBFS support for inheriting the IP TTL into the MPLS TTL at the ingress node by enabling TTL propagation. It ensures traceroute visibility across the MPLS network.

By default, the IP TTL is not inherited at the MPLS ingress node and the MPLS TTL is set to the value '64'.

**Supported Platforms**

Not all features are necessarily supported on each hardware platform. Refer to the Platform Guide for the features and the sub-features that are or are not supported by each platform.

## 6.1.2. Interfaces Configuration

**Configuration Hierarchy**

The diagram illustrates the interface configuration hierarchy.

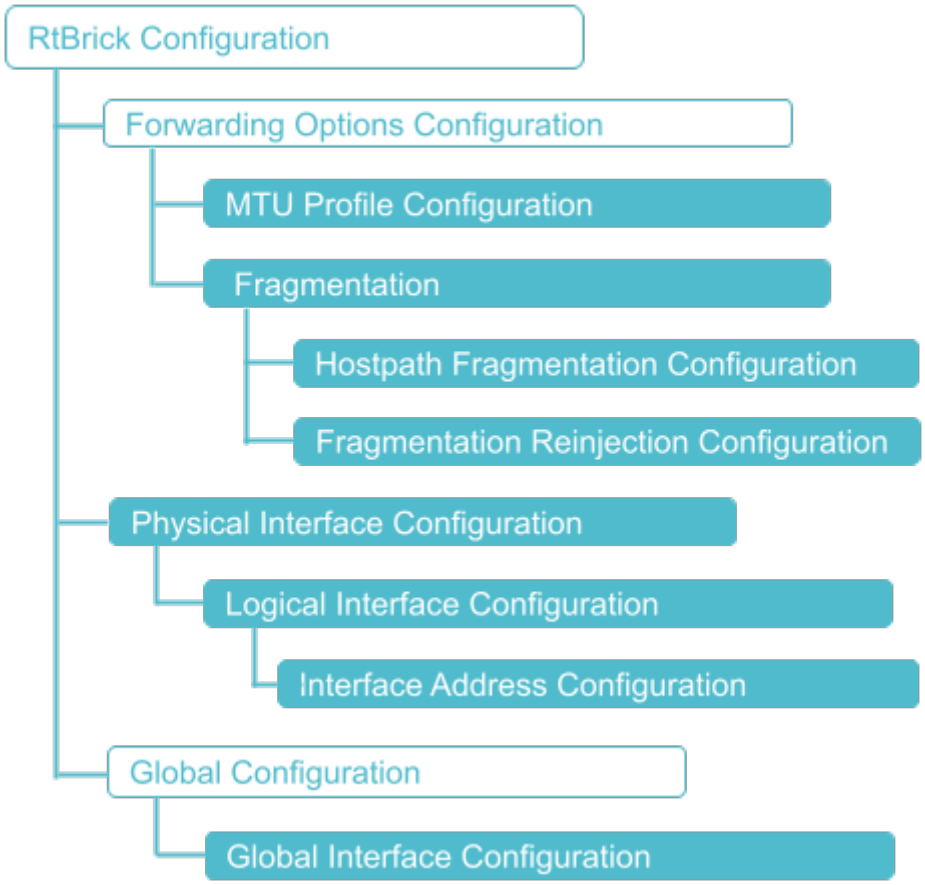



*Figure 113. Interface Configuration Hierarchy*

**Configuration Syntax and Commands**

The following sections describe the interface configuration syntax and commands.

**MTU Profile Configuration**

This section describes how to configure MTU profiles.

**Syntax:**

**set forwarding-options mtu-profile** <attribute> <value>

| Attribute | Description |
| --- | --- |
| `mtu-profile <mtu-profile-name>` | MTU profile name |
| `mtu-size <mtu-size>` | MTU size. Range: 64 to 9216 bytes |
| `mtu-type <mtu-type>` | Specify the MTU type:<br>• **physical**: Port based MTU profile<br>• **pppoe**: subscriber IFL-based MTU profile for L2TP and PPPoE. This MTU profile is used by PPPoE subscribers to set the default MTU size of 1492. A configured size of 1492 bytes limits the size of the IPv4 or IPv6 header plus payload.<br>• **ipv4**: MTU profile of type IPv4. Only IPv4 traffic on the logical interface will be impacted.<br>• **ipv6**: MTU profile of type IPv6. Only IPv6 traffic on the logical interface will be impacted.<br>• **ip**: MTU profile of type IP. Both IPv4 and IPv6 traffic on the logical interface will be impacted. |
| action <mtu-action> | Specify the MTU action. The following options are supported:<br>**drop**: This indicates that when the MTU check fails, the action "drop" is performed.<br>**redirect-to-cpu**: This is an action of redirecting packets to the CPU in a traffic behavior. A `redirect-to-cpu` action must be configured for fragmentation to occur. |

Example: Configuration of the MTU Profile for the Physical Port

```
{
  "ietf-restconf:data": {
    "rtbrick-config:forwarding-options": {
      "mtu-profile": [
        {
          "mtu-profile-name": "portMtu",
          "size": 5000,
          "type": "physical",
          "action": "redirect-to-cpu"
        }
      ]
    }
  }
}
```

Example 2: MTU Profile Configuration of Type IPv4

```
{
  "ietf-restconf:data": {
```

```
    "rtbrick-config:forwarding-options": {
      "mtu-profile": [
        {
          "mtu-profile-name": "ipv4Mtu",
          "size": 1300,
          "type": "ipv4",
          "action": "redirect-to-cpu"
        }
      ]
    }
  }
}
```

Example 3: MTU Profile Configuration of Type IPv6

```
{
  "ietf-restconf:data": {
    "rtbrick-config:forwarding-options": {
      "mtu-profile": [
        {
          "mtu-profile-name": "ipv6Mtu",
          "size": 1400,
          "type": "ipv6",
          "action": "redirect-to-cpu"
        }
      ]
    }
  }
}
```

Example 4: Configuration of the MTU Profile for PPPoE

```
{
  "ietf-restconf:data": {
    "rtbrick-config:forwarding-options": {
      "mtu-profile": [
        {
          "mtu-profile-name": "pppoeMtu",
          "size": 1492,
          "type": "pppoe",
          "action": "redirect-to-cpu"
        }
      ]
    }
  }
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Enabling Hostpath Fragmentation

This section describes how to enable or disable fragmentation by CPU. It is necessary to configure MTU profile action "redirect-to-cpu" so that fragmentation takes place. By default, fragmentation is disabled.

**Syntax:**

**set forwarding-options fragmentation ipv4 state** <value>

| Attribute | Description |
|---|---|
| `disabled \| cpu` | Enables fragmentation of IPv4 packets.<br>There are two options:<br>disabled—Fragmentation is disabled. It is the default setting.<br>cpu—Fragmentation is performed by CPU. |

Example: Configuration of Hostpath Fragmentation

```
{
  "ietf-restconf:data": {
    "rtbrick-config:forwarding-options": {
      "fragmentation": {
        "ipv4": {
          "state": "cpu"
        }
      }
    }
  }
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Configuring Policer Limits for Fragmented Traffic

RBFS optimizes performance for fragmented traffic by adjusting the default policer limits on the host path. Fragmentation packets received on the host path has a cap upto 150 Mbps, which enables higher throughput for fragmented traffic

**Syntax:**

**set forwarding-options fragmentation ipv4 state** <value>

**set forwarding-options fragmentation ipv4 policer** <policer-name>

The following policer configuration sets the RBFS host path to support 120 Mbps.

```
supervisor@rtbrick>SPINE01: cfg> set forwarding-options fragmentation ipv4 state cpu
supervisor@rtbrick>SPINE01: cfg> set forwarding-options fragmentation ipv4 policer
policer_120MB
```

**Example:**

```
supervisor@rtbrick>: cfg> show config forwarding-options fragmentation
{
  "rtbrick-config:fragmentation": {
    "ipv4": {
    "state": "cpu",
    "policer": "policer_120MB"
    }
  }
}
```

- RBFS supports default policers, which can be applied to fragmentation traffic through the policer configuration.
- In addition to the default policer, users can define a custom policer and attach it to the fragmentation configuration. However, the committed information rate (CIR) cannot exceed 150Mbps.

#### Physical Interface Configuration

This section describes configuration options at the physical interface (IFP) level.

**Syntax:**

**set interface** <interface-name> <attribute> <value>

| Attribute | Description |
|---|---|
| `<interface-name>` | Name of the interface. Example: ifp-0/0/1. |
| `admin-status <down\|up>` | Administrative state of the interface. |
| `auto-negotiation true` | Enable auto-negotiation. To disable auto-negotiation, use the `delete` form of the command.<br><br>NOTE: Port speed configuration and auto-negotiation are mutually exclusive. |
| `holddown <disable\|enable>` | Enable or disable the holddown timer feature. By default, the feature is enabled. |
| `holddown-down-delay <…>` | Configure the down delay time in milliseconds (ms). Default range 0 - 10000 ms. |

| Attribute | Description |
| --- | --- |
| `holddown-up-delay <…>` | Configure the down delay time in milliseconds (ms). Default range 0 - 10000 ms. |
| `class-of-service <profile-name>` | Apply class-of-service profile name. |
| `description` | Configure physical interface description. Maximum 1024 characters. |
| `host-if <container-interface>` | Configure a host interface. For example, if the container interface eth1 connects to the host interface vethXYZ123, use this command option to bound hostif-0/0/1 to eth1.<br><br>The Linux virtual ethernet (veth) interface needs to be created separately. It cannot be created via RBFS configuration. |
| `forward-error-correction <fec-type>` | Configure Forward Error Correction (FEC) on the physical interface. FEC allows you to send the required information to correct errors through the link along with the payload data. A benefit of "forward" in FEC is that errors detected at the receiver do not need to be retransmitted. Currently, the supported FEC types are: base-r, rsfec, none.<br><br>NOTE: `rsfec` is the only FEC supported for 100G on the QAX platform. |
| `link-training true` | Enable link training. To disable link training, use the `delete` form of the command. |
| `master <true\|false>` | Memif role, master or slave, applicable only to memif interface. One end needs to be configured as master, and the other one as slave. |
| `memif-id <id>` | Configure memif ID, applicable only to memif interface. Needs to match on both ends. |
| `mtu-profile <mtu-profile-name>` | Attach MTU profile to a physical interface. This is a mandatory attribute. |
| `mru <size>` | Maximum receive unit size on the physical interface. |
| `forward-error-correction-bypass <disable \| enable>` | When specify 'enable', the interface operates without applying FEC and reduces the processing latency. When disabled, the interface uses standard FEC operation as defined. |

| Attribute | Description |
| --- | --- |
| `speed <speed>` | Configure speed mode for the interface. Port speed refers to the maximum amount of data transmitted. The speed value is specified in Gigabits per second (Gbps).<br><br>Currently, RBFS supports 10G and 100G ports, and you can make the following changes:<br><br>• 100G port speed can be changed to 40G<br>• 10G port speed can be changed to 1G |

Example 1: Physical Interface Configuration

```
{
    "rtbrick-config:interface": [
      {
        "name": "ifp-0/0/1",
        "description": "Link to leaf1",
        "speed": "10G",
        "mtu-profile": "portMtu",
        "mru": 5000
      }
    ]
}
```

Example 2: Memory Interface Configuration

A End:

```
{
    "rtbrick-config:interface": [
      {
        "name": "ifp-0/0/1",
        "description": "Master",
        "memif-id": 11,
        "master": "true",
      }
    ]
}
```

B End:

```
{
    "rtbrick-config:interface": [
      {
        "name": "ifp-0/0/1",
        "description": "Slave",
```

```
        "memif-id": 11,
        "master": "false",
      }
    ]
  }
```

Example 3: Host Interface Configuration

```
{
    "rtbrick-config:interface": [
        "name": "ifp-0/0/1",
        "description": "Represents eth1 as ifp-0/0/1 in RBFS",
        "host-if": "eth1",
    ]
}
```

Example 4: MRU Configuration for Physical Interface

```
{
    "rtbrick-config:interface": [
      {
        "name": "ifp-0/0/7",
        "mru": 5000
      }
    ]
  }
```

Example 5: FEC Configuration for Physical Interface

```
{
    "rtbrick-config:interface": [
      {
        "name": "ifp-0/0/40",
        "forward-error-correction": "base-r"
      }
    ]
  }
```

API To access the RESTCONF API that corresponds to this CLI, click here.

API To access the Operational State API that corresponds to this CLI, click here.

#### Logical Interface Configuration

This section describes configuration options at the logical interface (IFL) level.

**Syntax:**

**set interface** <interface-name> **unit** <unit-id> <attribute> <value>

| Attribute | Description |
|---|---|
| `unit <unit-id>` | Create a logical interface (also referred to as a sub-interface) under the physical interface. |
| `address` | Configure IPv4 and IPv6 address for sub-interface. |
| `address-translation` | Configure NAT for logical interface. |
| `admin-status <down\|up>` | Administrative state of the logical interface. |
| `class-of-service <profile-name>` | Apply class-of-service profile name. |
| `description <description>` | Description of the logical interface. Maximum 1024 characters. |
| `esi` | Configure Ethernet Segment Identifier for L2 interface. |
| `garp-disable true` | Disables Gratuitous ARP (GARP). Default is GARP enabled. |
| `inner-vlan <inner-vlan-id>` | Inner VLAN ID. |
| `input-vlan-map` | Configure input VLAN mapping |
| `input-vlan-map inner-vlan-id <inner-vlan-id>` | Configure the inner VLAN for the VLAN operation specified |
| `input-vlan-map vlan-encapsulation <802.1ad \| 802.1q>` | VLAN encapsulation at ingress |
| `input-vlan-map vlan-id <vlan-id>` | Configure the vlan for the vlan operation specified |
| `input-vlan-map vlan-operation <push \| pop \| swap \| push-push \| pop-pop \| swap-swap \| swap-push \| pop-swap>` | VLAN operation at ingress |
| `instance <instance>` | Assign the logical interface to an instance. |
| `interface-type` | Configure L2 interface type |
| `ip-mtu-profile <ip-mtu-profile>` | Attach IP MTU profile to an L3 interface. |
| `ipv4-admin-status <down\|up>` | Enable or disable IPv4. |
| `ipv4-mtu-profile <ipv4-mtu-profile>` | Attach IPv4 MTU profile to an L3 interface. |
| `ipv6-admin-status <down\|up>` | Enable or disable IPv6. |
| `ipv6-mtu-profile <ipv6-mtu-profile>` | Attach IPv6 MTU profile to an L3 interface. |
| `ip-mtu-profile <ip-mtu-profile>` | Attach IP MTU profile to an L3 interface. |
| `mpls-admin-status <down \| up>` | Enable or disable MPLS. |
| `mpls-mtu <mpls-mtu-size>` | MPLS maximum transmission unit size. |

| Attribute | Description |
|---|---|
| `neighbor <ipv4 \| ipv6> <ip-address> mac <mac-address>` | Configure a static IPv4 or IPv6 neighbor. |
| `output-vlan-map` | Configure Output VLAN mapping |
| `output-vlan-map vlan-operation` | Specifies the VLAN operation at egress |
| `output-vlan-map vlan-priority-operation` | Specifies the VLAN priority operation, which allows users to configure the outer-VLAN-priority bits to remain transparent. |
| `output-vlan-map inner-vlan-id <inner-vlan-id>` | Configure the inner VLAN for the VLAN operation specified |
| `output-vlan-map vlan-encapsulation <802.1ad \| 802.1q>` | VLAN encapsulation at egress |
| `output-vlan-map vlan-id <vlan-id>` | Configure the VLAN for the VLAN operation specified |
| `output-vlan-map vlan-operation <push \| pop \| swap \| push-push \| pop-pop \| swap-swap \| swap-push \| pop-swap>` | VLAN operation at egress |
| `unnumbered interface <loopback-interface-name>` | Configure an unnumbered interface. See the Unnumbered Interface Configuration example below. |
| `ra-disable true` | Disable the router advertisement functionality. |
| `service-meta-data` | Configure layer-2 service egress meta data value. |
| `statistics <false \| true>` | Enable traffic statistics for the L2 logical interfaces (L2 IFLs). For L3 logical interfaces (L3 IFLs), traffic statistics are enabled by default. The value `true` indicates enabled, and the value `false` indicates disabled. |
| `vlan <outer-vlan-id>` | Specifies Outer VLAN ID. |
| `vlan-encapsulation <802.1ad \| 802.1q>` | Specifies the outer VLAN tag. |

Example 1: Logical Interface Configuration with IPv4 MTU Profile

```
{
  "rtbrick-config:interface": [
     {
       "name": "ifp-0/0/1",
       "unit": [
         {
           "unit-id": 1,
           "description": "VLAN 101",
           "instance": "default",
           "ipv4-mtu-profile": "ipv4Mtu"

         }
       ]
```

```
      }
    ]
  }
```

Example 2: Logical Interface Configuration with IPv6 MTU Profile

```
  {
    "rtbrick-config:interface": [
      {
        "name": "ifp-0/0/1",
        "unit": [
          {
            "unit-id": 1,
            "description": "VLAN 101",
            "instance": "default",
            "ipv6-mtu-profile": "ipv6Mtu"

          }
        ]
      }
    ]
  }
```

Example 3: Logical Interface Configuration with IP MTU Profile

```
  {
    "rtbrick-config:interface": [
      {
        "name": "ifp-0/0/1",
        "unit": [
          {
            "unit-id": 1,
            "description": "VLAN 101",
            "instance": "default",
            "ip-mtu-profile": "ipMtu"

          }
        ]
      }
    ]
  }
```

Example 4: Logical Interface Configuration with disabling router advertisement

```
  {
    "ietf-restconf:data": {
      "rtbrick-config:interface": [
```

```
      {
        "name": "ifp-0/0/1",
        "ifp": "L1-4-S1"
      },
      {
        "name": "hostif-0/0/2",
        "ifp": "L1-5-SN",
        "unit": [
          {
            "unit-id": 10,
            "ra-disable": "true"
          }
        ]
      },
    ]
  }
}
```

Example 5: Unnumbered Interface Configuration

The following configuration sets up a loopback interface `lo-0/0/0/0` with IP addresses, and creates two unnumbered interfaces that inherits or borrows the IP addresses from the loopback interface.

```
supervisor@rtbrick: cfg> set interface lo-0/0/0 unit 0 instance service
supervisor@rtbrick: cfg> set interface lo-0/0/0 unit 0 address ipv4 192.0.2.1/32
supervisor@rtbrick: cfg> set interface lo-0/0/0 unit 0 address ipv6 2001:db8::1/128

supervisor@rtbrick: cfg> set interface ifp-0/0/25 unit 0 unnumbered interface lo-
0/0/0/0
supervisor@rtbrick: cfg> set interface ifp-0/0/25 unit 1 unnumbered interface lo-
0/0/0/0
supervisor@rtbrick: cfg> commit
```

```
supervisor@rtbrick: cfg> #show config interface#
{
  "rtbrick-config:interface": [
    {
      "name": "ifp-0/0/25",
      "unit": [
        {
          "unit-id": 0,
          "instance": "service",
          "vlan": 20,
          "unnumbered": {
            "interface": "lo-0/0/0/0"
          }
        },
        {
          "unit-id": 1,
```

```
          "instance": "service",
          "vlan": 21,
          "unnumbered": {
            "interface": "lo-0/0/0/0"
          }
        }
      ]
    },
    {
      "name": "lo-0/0/0",
      "unit": [
        {
          "unit-id": 0,
          "instance": "service",
          "address": {
            "ipv4": [
              {
                "prefix4": "192.168.1.1/32"
              }
            ],
            "ipv6": [
              {
                "prefix6": "192:168:1::1/128"
              }
            ]
          }
        }
      ]
    }
  ]
}
```

Example 6: The following example configures the interface (`ifp-0/1/10 unit 10`) to ignore outer-VLAN-priority rewrite operations by setting the `vlan-priority-operation` to `none` within the `output-vlan-map` configuration.

```
supervisor@rtbrick>SPINE01: cfg> set interface ifp-0/1/10 unit 10 output-vlan-map
supervisor@rtbrick>SPINE01: cfg> set interface ifp-0/1/10 unit 10 output-vlan-map
vlan-priority-operation none
```

```
 {
   "rtbrick-config:interface": [
      {
        "name": "ifp-0/0/1",
        "unit": [
          {
            "unit-id": 1,
            "description": "VLAN 101",
```

```
            "instance": "default",
            "output-vlan-map":
              "vlan-priority-operation": none
          }
        ]
      }
    ]
  }
```

To access the RESTCONF API that corresponds to this CLI, click here.

#### Configuring IPv6 and MPLS Address Families on Logical Interfaces

You can explicitly enable IPv6 and MPLS address families on the required logical interfaces. Use the following configuration commands to enable IPv6 and MPLS address families on logical interfaces.

**set interface** <interface-name> **unit** <unit id> **ipv6-admin-status up**
**set interface** <interface-name> **unit** <unit id> **mpls-admin-status up**

| Attribute | Description |
|---|---|
| `<interface-name>` | Name of the interface. |
| `unit <unit-id>` | Create a logical interface (also referred to as a sub-interface) under the physical interface. |
| `ipv6-admin-status up` | Enable IPv6 protocol processing on the interface unit. |
| `mpls-admin-status up` | Enables IPv6 MPLS protocol processing on the interface unit. |

This configuration example below enables IPv6 and MPLS address families on multiple logical units of interface ifp-0/0/16. The 'ipv6-admin-status' and 'mpls-admin-status' options are specified as 'up', indicating that the address families are enabled in the specified units of the interface ifp-0/0/16. Each unit is mapped to a separate routing instance and assigned IPv4 and/or IPv6 addresses.

```
set interface ifp-0/0/16 speed 10G
set interface ifp-0/0/16 unit 1 ipv6-admin-status up
set interface ifp-0/0/16 unit 1 mpls-admin-status up
set interface ifp-0/0/16 unit 1 instance radius-vrf
set interface ifp-0/0/16 unit 1 vlan 100
set interface ifp-0/0/16 unit 1 address ipv4 192.0.2.1/24
set interface ifp-0/0/16 unit 10 ipv6-admin-status up
set interface ifp-0/0/16 unit 10 mpls-admin-status up
set interface ifp-0/0/16 unit 10 instance vrf1
set interface ifp-0/0/16 unit 10 address ipv4 192.0.2.2/24
set interface ifp-0/0/16 unit 10 address ipv6 2001:db8:0:0:1:0:0:1/64
```

```
supervisor@rtbrick.net: op> show config interface ifp-0/0/16
```

```
{
  "rtbrick-config:interface": [
    {
      "name": "ifp-0/0/16",
      "speed": "10G",
      "unit": [
        {
          "unit-id": 1,
          "ipv6-admin-status": "up",
          "mpls-admin-status": "up",
          "instance": "radius-vrf",
          "vlan": 100,
          "address": {
            "ipv4": [
              {
                "prefix4": "192.0.2.1/24"
              }
            ]
          }
        },
        {
          "unit-id": 10,
          "ipv6-admin-status": "up",
          "mpls-admin-status": "up",
          "instance": "vrf1",
          "address": {
            "ipv4": [
              {
                "prefix4": "192.0.2.2/24"
              }
            ],
            "ipv6": [
              {
                "prefix6": "2001:db8:0:0:1:0:0:1/64"
              }
            ]
          }
        }
      ]
    }
  ]
}
```

**Interface Address Configuration**

This section describes how to configure interface IP addresses.

**Syntax:**

**set interface** <interface-name> **unit** <unit-id> **address** <afi> <attribute> <value>

| Attribute | Description |
|---|---|
| `<afi>` | Address family identifier (AFI). Supported values: ipv4 and ipv6 |
| `<prefix4\|prefix6>` | Assign IPv4 or IPv6 address to the interface unit. |
| `community <community-value>` | Configure list of communities associated with the address. |
| `extended-community <community-value>` | Configure list of extended communities associated with the address. |
| `label <label-value>` | Configure label associated with the address.<br>Supported MPLS label values are 0 - 1048575. The reserved MPLS label range is 0 - 15. In RBFS, BGP uses the label range 20000 - 100000. It is recommended to assign label values outside of these reserved ranges to avoid conflicts. |
| `secondary true` | Enable a secondary IPv4 address. To disable the `secondary` IP configuration, use the `delete` form of the command. |

Broadcast and network IP addresses cannot be configured on logical (IFL) interfaces.

Example: Interface Address Configuration

```
{
    "rtbrick-config:interface": [
      {
        "name": "lo-0/0/1",
        "unit": [
          {
            "unit-id": 1,
            "address": {
              "ipv4": [
                {
                  "prefix4": "198.51.100.103/24",
                  "label": 12346
                }
              ]
            }
          }
        ]
      }
    ]
  }
```

To access the RESTCONF API that corresponds to this CLI, click here.

#### Global Interface Configuration

This section describes a configuration option applied globally to all interfaces.

**Syntax:**

**set global interface all** <attribute> <value>

| Attribute | Description |
|---|---|
| `admin-status <up\|down>` | Configure state of the interface. |

- The interface level enable/disable command has higher precedence than the global interface enable/disable command.
- You can disable all unused physical interfaces.
- Before executing the global interface disable all command ensure that all physical interfaces are in the link Up state.

Example: Enabling or Disabling all Interfaces

```
{
  "ietf-restconf:data": {
    "rtbrick-config:global": {
      "interface": {
        "all": {
          "admin-status": "down"
        }
      }
    }
  }
}
```

#### Interface Holddown Configuration

By default, the Interface Hold-down feature is enabled. You can disable the feature if you want. It is allowed to configure timer value for options such as 'holddown down delay' and 'holddown up delay'.

**Syntax:**

**set interface** <interface-name> <attribute> <value>

| Attribute | Description |
|---|---|
| `<interface-name>` | Name of the interface. Example: ifp-0/0/1. |

| Attribute | Description |
|---|---|
| `holddown <disable\|enable>` | Enable or disable the holddown timer feature. By default, the feature is enabled. |
| `holddown-down-delay` | Configure the down delay time in milliseconds (ms). The allowed values include 0, 100, 200, 500, 1000, 2000, 5000, and 10000. Default 0 ms. |
| `holddown-up-delay` | Configure the up delay time milliseconds (ms). The allowed values include 0, 100, 200, 500, 1000, 2000, 5000, and 10000. Default range 0 - 10000 ms. |

#### IPv6 Link-Local Address Configuration

You can manually assign an IPv6 link-local address on an interface.

**Syntax:**

**set interface** <interface-name> **unit** <unit-id> **address ipv6-link-local** <IPv6 address>

| Attribute | Description |
|---|---|
| `<interface-name>` | Name of the interface. Example: ifp-0/0/1. |
| `unit <unit-id>` | Create a logical interface (also referred to as a sub-interface) under the physical interface. |
| `ipv6-link-local <IPv6 address>` | Specifies the IPv6 link-local address. |

Example for IPv6 link-local configuration

```
{
 "ietf-restconf:data": {
   "rtbrick-config:interface": [
     {
       "name": "hostif-0/0/0",
       "host-if": "B1-1-SN"
     },
     {
       "name": "hostif-0/0/2",
       "host-if": "B1-3-B2",
       "unit": [
         {
           "unit-id": 10,
           "address": {
             "ipv6-link-local": "fe80::1"
           }
         }
       ]
     },
     {
       "name": "hostif-0/0/3",
```

```
        "host-if": "B1-4-B2"
      }
    ]
  }
}
```

#### Disabling Gratuitous ARP on Interface

You can disable the sending of Gratuitous ARP (GARP) on an interface, which is enabled by default on all interfaces.

> To enable GARP on an interface, you need to delete the `garp-disable true` configuration.

**Syntax:**

**set interface** <interface-name> **unit** <unit-id> **garp-disable true**

| Attribute | Description |
|---|---|
| interface <interface-name> | Name of the interface. Examples: ifp-0/0/1. |
| garp-disable true | Disables Gratuitous ARP (GARP). Default is GARP enabled. |

Example: Disabling Gratuitous ARP

```
supervisor@rtbrick.net: cfg> set interface ifp-0/0/0 unit 100 garp-disable true
```

#### Static ARP Entry Configuration on Logical Interfaces

This static ARP Entry on IFLs enables the RBFS devices to proactively resolve ARP and ND.

**Syntax:**

**set interface <interface> unit** <unit id> **neighbor** < **ipv4** | **ipv6** > <ip address> **dynamic true**

The following example shows how to configure an ARP entry on a logical interface.

```
set interface ifp-0/1/0 unit 6 neighbor ipv4 192.0.2.1 dynamic true
```

**Example:**

```
{
  "ietf-restconf:data": {
    "rtbrick-config:interface": [
      {
        "name": "ifp-0/1/0",
```

```
        "host-if": "S1-1-SN",
        "unit": [
          {
            "unit-id": 10,
            "neighbor": {
              "ipv4": [
                {
                  "address": "192.0.2.1",
                  "dynamic": "true"
                }
              ]
            }
          }
        ]
      },
      {
        "name": "ifp-0/0/1",
        "host-if": "S1-2-SN"
      },
      {
        "name": "ifp-0/0/2",
        "host-if": "S1-3-L1"
      },
      {
        "name": "ifp-0/0/3",
        "host-if": "S1-4-L1"
      }
    ]
  }
}
```

To access the RESTCONF API that corresponds to this CLI, click here.

**Breakout Interfaces**

RBFS supports breakout for 100G interfaces on the platforms: UfiSpace S9610-46DX, UfiSpace S9600-32X, and EdgeCore AGR400 (AS7946-30XB). Using the breakout configuration, the high-speed interfaces(100G) on these platforms can be divided into multiple lower-speed interfaces(10G or 25G), allowing them to connect with lower rate ports(10G or 25G) in other network devices.

For more information about UfiSpace S9610-46DX breakout interfaces, see Interface Breakout Support on UfiSpace S9610-46DX.

RBFS supports the following breakout configuration options:

| Rate | Technology | Breakout Capable | Electric Lanes | Optical Lanes |
|---|---|---|---|---|
| 100G | QSFP28 | Yes | 4 x 25G | 4 x 25G |
| 100G | QSFP28 | Yes | 4 x 10G | 4 x 10G |

For example, to set up the individual peer-side (25G/10G) interfaces, we need to configure the breakout on the 100G port to 4x10G or 4x25G based on the peer-side speed.

When a breakout is configured on a high-speed interface (for example, ifp-0/1/12), it is divided into four lower-speed interfaces: ifp-0/1/12#0, ifp-0/1/12#1, ifp-0/1/12#2, and ifp-0/1/12#3. As a result, the original high-speed interface will no longer be available.

The following diagram shows the breakout for the interface ifp-0/1/12 on the high-speed 100G to 4 x 25G ports (0, 1, 2, 3).

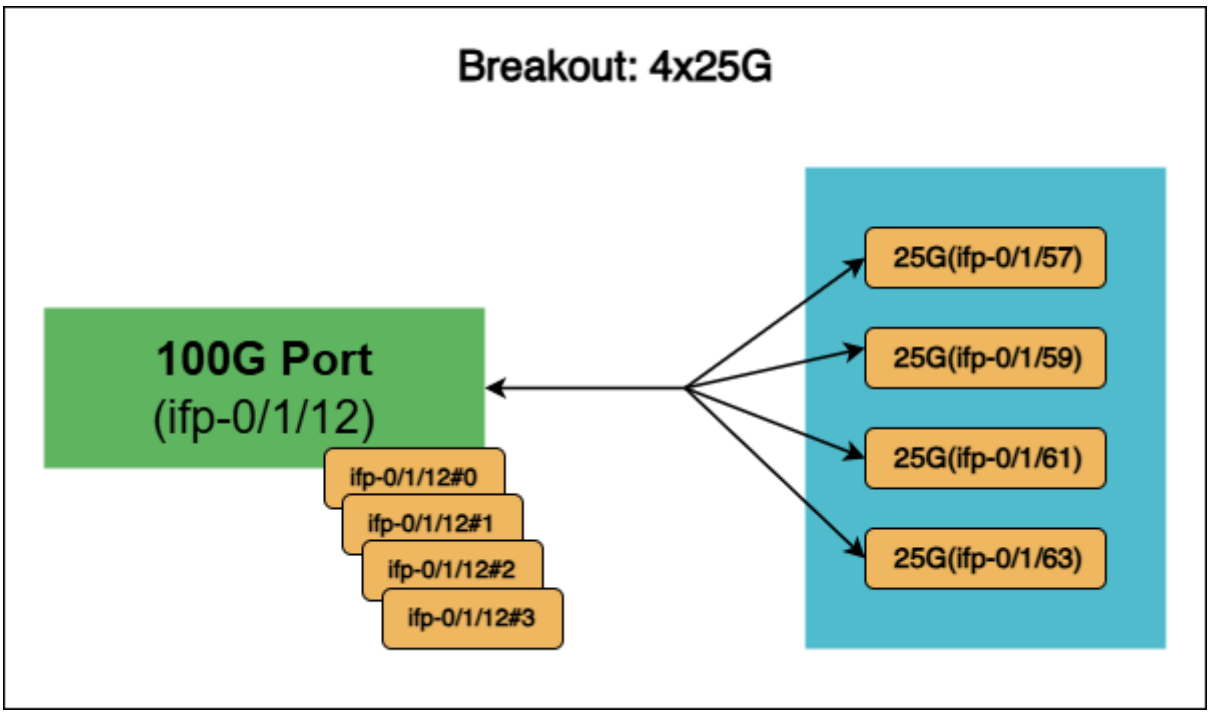



*Figure 114. Breakout Configuration*

#### Guidelines and Limitations

Currently, the following limitations exist.

- Autonegotiation and traffic sampling are not supported for breakout interfaces.
- A breakout interface cannot be a member of a LAG.
- Using breakout interfaces with routing protocols is not supported.

#### Configuring Breakout

To configure for the interface ifp-0/1/12 on the high-speed 100G to 4x25G ports., enter the following command:

**Syntax:**

```
set interface <interface-name> breakout map <4x10G | 4x25G>
```

**Example:**

```
set interface ifp-0/1/12 breakout map 4x25G
```

#### Configuring Speed on the Breakout Interfaces

RBFS supports for configuring speed on breakout interfaces. You can explicitly configure the speed on individual breakout interfaces.

**Syntax:**

**set interface** <breakout-ifp-name> **speed** <speed>

| Option | Description |
|---|---|
| `<breakout-ifp-name>` | Specifies the breakout interface on which the speed is to be configured. |
| `<speed>` | Configures the operational speed for the specified breakout interface. |

Example:

The command configures the operational speed '10G' for the breakout logical interface (ifp-0/1/12#0).

```
set interface ifp-0/1/12#0 speed 10G
```

```
{
  "interface": {
    "ifp-0/1/12#0": {
      "speed": 10G"
    }
  }
}
```

#### Split Horizon for P2MP Interface

Split Horizon for P2MP Interface improves security and traffic control by preventing direct communication between devices connected to the same Point-to-Multipoint (P2MP) interface. Currently, in a P2MP setup where multiple devices (like OLTs) are connected to a single interface, these devices can communicate directly with each other. This is often undesirable for security and network management.

Therefore, a `no-local-routing` configuration knob for P2MP interfaces is made available. When enabled, this feature will block direct communication between hosts connected to the same P2MP interface.

When `no-local-routing true` is configured on an interface (e.g., a loopback interface associated with the P2MP interface), the system will automatically create Access Control Lists (ACLs) to:

- **Block**: Traffic originating from the P2MP subnet and destined for the same P2MP subnet. This prevents communication between devices on that P2MP interface.
- **Allow**: Traffic from the P2MP subnet to the interface's own IP address, allowing traffic to the gateway from the client.

Enabling split horizon provides these key advantages:

- **Improved Security**: Prevents unauthorized communication between devices within the same P2MP segment, isolating them from each other.
- **Enhanced Network Control**: Gives administrators granular control over traffic flow on P2MP interfaces.
- **Compliance**: Helps meet security requirements that mandate isolation between network segments or connected devices.
- **Simplified Configuration**: A single configuration knob, `no-local-routing true`, simplifies the process of implementing split horizon, as the system automatically manages the underlying ACLs.

This feature is supported on all platforms except the QAX platform.

#### Configuring Split Horizon

The Split Horizon for P2MP Interface feature enhances network security and control for Point-to-Multipoint (P2MP) interfaces.

Syntax:

**set interface** <interface-name> **unit** <unit-number> **address** <address> **no-local-routing true**

| Attribute | Description |
|---|---|
| `interface-name` | The name of the interface to configure (e.g., `lo-0/0/6` for a loopback interface). |
| `unit-number` | The unit number of the interface (e.g., `4`). |
| `address` | The IP address to assign to the interface (e.g., `198.51.100.1/24`). |
| `address no-local-routing true` | Enables split horizon by preventing local routing on the interface, blocking traffic between devices on the same P2MP interface. |

The example below configures a loopback interface (`lo-0/0/6 unit 4`) with an IPv4 address and enables `no-local-routing true` for split horizon. It also configures a VLAN interface (`ifp-0/1/14 unit 4`) as unnumbered, inheriting its IP from the loopback, to separate management and in-band traffic and prevent routing loops.

```
set interface lo-0/0/6 unit 4 description "lo0.4 dhcp_relay"
set interface lo-0/0/6 unit 4 instance oob_mgmt
set interface lo-0/0/6 unit 4 address ipv4 198.51.100.1/24 no-local-routing true
set interface ifp-0/1/14 unit 4 description "vlan_inband"
set interface ifp-0/1/14 unit 4 instance oob_mgmt
set interface ifp-0/1/14 unit 4 vlan 4
set interface ifp-0/1/14 unit 4 unnumbered interface lo-0/0/6/4
set interface ifp-0/1/14 unit 4 garp-disable true
```


```
{
  "rtbrick-config:interface": [
    {
      "name": "ifp-0/1/14",
      "unit": [
        {
          "unit-id": 4,
          "description": "vlan_inband",
          "instance": "oob_mgmt",
          "vlan": 4,
          "unnumbered": {
            "interface": "lo-0/0/6/4"
          },
          "garp-disable": "true"
        }
      ]
    },
    {
      "name": "lo-0/0/6",
      "unit": [
        {
          "unit-id": 4,
          "description": "lo0.4 dhcp_relay",
          "instance": "oob_mgmt",
          "address": {
            "ipv4": [
              {
                "prefix4": "198.51.100.1/24",
                "no-local-routing": "true"
              }
            ]
          }
        }
      ]
    }
  ]
}
```

#### Support for FEC Bypass

RBFS supports Forward Error Correction (FEC) bypass on high-speed Ethernet interfaces. It allows low-latency operation by enabling FEC to be bypassed while preserving link compatibility. In latency-sensitive environments such as high-frequency trading, disabling FEC helps to reduce end-to-end latency. FEC bypass requires coordination with the link partner (at the other end), as the receiver must be aware that FEC is not applied.

When enabled, the transmitter explicitly signals the FEC bypass status to the link partner, preventing link failures caused by FEC mismatches.

**Syntax:**

**set interface** <interface-name> **forward-error-correction-bypass** <disable | enable>

| Attribute | Description |
|---|---|
| `<interface-name>` | Name of the physical interface. |
| `forward-error-correction-bypass <disable \| enable>` | When specify 'enable', the interface operates without applying FEC and reduces the processing latency. When disabled, the interface uses standard FEC operation as defined. |

Example Command:

The following example configuration enables FEC bypass on the specified interface. It indicates that traffic on this interface will operate without FEC processing to support low-latency operation.

```
set interface ifp-0/1/8 forward-error-correction-bypass enable
```

```
{
  "rtbrick-config:interface": [
    {
      "name": "ifp-0/1/8",
      "speed": "10G",
      "forward-error-correction-bypass": "enable"
    }
  ]
}
```

#### Enabling TTL Inheritance from IP to MPLS

You can enable the inheritance of the IP TTL into the MPLS TTL at the ingress node by enabling TTL propagation.

By default, the IP TTL is not inherited at the MPLS ingress node and the MPLS TTL is set to the value '64'. To enable this feature, use the following configuration

command:

**Syntax:**

**set forwarding-options ttl propagate enable**

| Attribute | Description |
|---|---|
| `ttl propagate enable` | Enables the router to inherit the incoming packet's IP TTL to the outgoing packet's MPLS TTL. |

Example:

The following configuration allows the device to inherit the incoming IP TTL/hop-limit into the MPLS TTL.

```
set forwarding-options ttl propagate enable
```

```
{
  "rtbrick-config:ttl": {
    "propagate": "enable"
  }
}
```

#### Enabling Reverse Path Forwarding

This configuration enbles Reverse Path Forwarding (RPF) on an interface unit. Packets received on IFLs can be validated against their expected reverse path.

**Syntax**:

**set interface** <name> **unit** <unit-id> **rpf** [**loose** | **strict**]

| Attribute |
|---|
| Description |
| `<unit-id>` |
| Logical interface number. |
| `strict` |
| Accepts the incoming packet only if the source IP address is reachable through the same interface on which the packet is received. |
| `loose` |

| |
|---|
| Accepts the incoming packet if the source IP address exists in the routing table, regardless of the incoming interface. |

The following configuration enables Reverse Path Forwarding on the specified interface unit to validate the source of incoming IP packets. This helps prevent IP spoofing by ensuring that packets are received on a valid and reachable path, based on the configured validation mode, that is 'strict'.

```
set interface ifp-0/0/16 unit 100 rpf strict
```

```
{
  "interfaces": [
    {
      "name": "ifp-0/0/16",
      "unit": [
        {
          "unit-id": 100,
          "rpf": {
            "mode": "strict"
          }
        }
      ]
    }
  ]
}
```

### 6.1.3. Interfaces Operational Commands

#### Interface Show Commands

The interface show commands provide detailed information about the status and parameters of RBFS interfaces.

##### Interface Summary Commands

**Syntax:**

**show interface** <option>

| Option | Description |
|---|---|
| `summary` | Displays a summary of all interfaces including physical, logical, and address information. |
| `<interface-name>` | Displays a summary of an interface including physical, logical, and address information. |

| Option | Description |
|---|---|
| physical | Displays all physical interface including loopback, cpu and recycle ports. |
| logical | Displays all logical interfaces for all instances. |
| logical <instance-name> | Displays all logical interfaces for the given instance. |
| address | Displays all IPv4 and IPv6 addresses for all instances. |
| address <instance-name> | Displays all IPv4 and IPv6 addresses for the given instance. |

Example 1: Summary Output for All Interfaces

```
supervisor@rtbrick: op> show interface summary
Interface                Admin    Link     Oper          IPv4 Address             IPv6 Address
ifp-0/1/0                Up       Up       Up
ifp-0/1/1                Up       Down     Down
ifp-0/1/2                Up       Up       Up
ifp-0/1/3                Up       Up       Up
ifp-0/1/4                Up       Down     Down
ifp-0/1/5                Up       Up       Up
ifp-0/1/6                Up       Up       Up
ifp-0/1/7                Up       Up       Up
ifp-0/1/8                Up       Up       Up
ifp-0/1/9                Up       Down     Down
ifp-0/1/10               Up       Up       Up
ifp-0/1/11               Up       Down     Down
ifp-0/1/12               Up       Up       Up
ifp-0/1/13               Up       Down     Down
ifp-0/1/14               Up       Up       Up
ifp-0/1/15               Up       Down     Down
ifp-0/1/16               Up       Up       Up
ifp-0/1/17               Up       Down     Down
ifp-0/1/18               Up       Down     Down
ifp-0/1/19               Up       Down     Down
ifp-0/1/20               Up       Up       Up
ifp-0/1/21               Up       Up       Up
ifp-0/1/22               Up       Up       Up
  ifl-0/1/22/0           Up       Up       Up            192.0.2.2/30             fe80::5c07:58ff:fe79:a17/64
  ifl-0/1/22/100         Up       Up       Up            192.0.2.6/30             fe80::5c07:5800:6479:a17/64
  ifl-0/1/22/200         Up       Up       Up            192.0.2.10/30            fe80::5c07:5800:c879:a17/64
  ifl-0/1/22/300         Up       Up       Up            192.0.2.14/30            fe80::5c07:5801:2c79:a17/64
ifp-0/1/23               Up       Down     Down
ifp-0/1/24               Up       Up       Up
ifp-0/1/25               Up       Up       Up
ifp-0/1/26               Up       Up       Up
ifp-0/1/27               Up       Down     Down
ifp-0/1/28               Up       Down     Down
ifp-0/1/29               Up       Down     Down
ifp-0/1/30               Up       Down     Down
ifp-0/1/31               Up       Down     Down
ifp-0/1/32               Up       Up       Up
ifp-0/1/33               Up       Up       Up
ifp-0/1/34               Up       Down     Down
ifp-0/1/35               Up       Down     Down
ifp-0/1/36               Up       Down     Down
ifp-0/1/37               Up       Down     Down
ifp-0/1/38               Up       Down     Down
```

```
ifp-0/1/39              Up       Down     Down
ifp-0/1/40              Up       Down     Down
ifp-0/1/41              Up       Up       Up
ifp-0/1/42              Up       Up       Up
ifp-0/1/43              Up       Down     Down
ifp-0/1/44              Up       Down     Down
ifp-0/1/45              Up       Down     Down
ifp-0/1/46              Up       Up       Up
ifp-0/1/47              Up       Up       Up
ifp-0/1/48              Up       Up       Up
ifp-0/1/49              Up       Up       Up
ifp-0/1/50              Up       Down     Down
ifp-0/1/51              Up       Down     Down
ifp-0/1/52              Up       Down     Down
ifp-0/1/53              Up       Down     Down
ifp-0/1/54              Up       Down     Down
ifp-0/1/55              Up       Down     Down
ifp-0/1/56              Up       Down     Down
ifp-0/1/57              Up       Down     Down
ifp-0/1/58              Up       Down     Down
ifp-0/1/59              Up       Down     Down
ifp-0/1/60              Up       Down     Down
ifp-0/1/61              Up       Down     Down
ifp-0/1/62              Up       Down     Down
ifp-0/1/63              Up       Down     Down
ifp-0/1/64              Up       Down     Down
ifp-0/1/65              Up       Down     Down
ifp-0/1/66              Up       Down     Down
ifp-0/1/67              Up       Down     Down
ifp-0/1/68              Up       Down     Down
ifp-0/1/69              Up       Down     Down
ifp-0/1/70              Up       Down     Down
ifp-0/1/71              Up       Down     Down
ifp-0/1/72              Up       Down     Down
ifp-0/1/73              Up       Down     Down
ifp-0/1/74              Up       Down     Down
ifp-0/1/75              Up       Down     Down
ifp-0/1/76              Up       Down     Down
ifp-0/1/77              Up       Down     Down
ifp-0/1/78              Up       Down     Down
ifp-0/1/79              Up       Down     Down
ifp-0/1/80              Up       Down     Down
ifp-0/1/81              Up       Down     Down
ifp-0/1/82              Up       Down     Down
ifp-0/1/83              Up       Down     Down
ifp-0/1/84              Up       Down     Down
ifp-0/1/85              Up       Down     Down
ifp-0/1/86              Up       Down     Down
ifp-0/1/87              Up       Down     Down
ifp-0/1/88              Up       Down     Down
ifp-0/1/89              Up       Down     Down
ifp-0/1/90              Up       Down     Down
ifp-0/1/91              Up       Down     Down
ifp-0/1/92              Up       Down     Down
ifp-0/1/93              Up       Down     Down
ifp-0/1/94              Up       Down     Down
ifp-0/1/95              Up       Down     Down
ifp-0/1/96              Up       Up       Up
ifp-0/1/97              Up       Up       Up
ifp-0/1/98              Up       Down     Down
ifp-0/1/99              Up       Down     Down
```

```
ifp-0/1/100              Up       Down     Down
ifp-0/1/101              Up       Up       Up
cpu-0/0/200              Up       Up       Up
cpu-0/0/201              Up       Down     Down
cpu-0/0/202              Up       Down     Down
cpu-0/0/203              Up       Down     Down
recycle-0/0/211          Up       Up       Up
recycle-0/0/212          Up       Up       Up
recycle-0/0/213          Up       Up       Up
recycle-0/0/214          Up       Up       Up
recycle-0/0/215          Up       Up       Up
recycle-0/0/216          Up       Up       Up
recycle-0/0/217          Up       Up       Up
```

### Example 2: Summary Output for One Physical Interface

```
supervisor@rtbrick: op> show interface ifp-0/1/22
Interface                Admin    Link     Oper          IPv4 Address             IPv6 Address
ifp-0/1/22               Up       Up       Up
  ifl-0/1/22/0           Up       Up       Up            192.0.2.2/30             fe80::5c07:58ff:fe79:a17/64
  ifl-0/1/22/100         Up       Up       Up            192.0.2.6/30             fe80::5c07:5800:6479:a17/64
  ifl-0/1/22/200         Up       Up       Up            192.0.2.10/30            fe80::5c07:5800:c879:a17/64
  ifl-0/1/22/300         Up       Up       Up            192.0.2.14/30            fe80::5c07:5801:2c79:a17/64
```

### Example 3: List of All Physical Interfaces

```
supervisor@rtbrick: op> show interface physical
Interface              Admin   Link    Oper    Damp      MAC Address          Speed  Duplex  Uptime
ifp-0/1/0              Up      Up      Up      stable    5c:07:58:79:0a:01    25G    Full    Fri Jun 05 12:37:40 GMT
+0000 2026
ifp-0/1/1              Up      Down    Down    stable    5c:07:58:79:0a:02    25G    Full    never
ifp-0/1/2              Up      Up      Up      stable    5c:07:58:79:0a:03    10G    Full    Fri Jun 05 12:37:40 GMT
+0000 2026
ifp-0/1/3              Up      Up      Up      stable    5c:07:58:79:0a:04    10G    Full    Fri Jun 05 12:37:40 GMT
+0000 2026
ifp-0/1/4              Up      Down    Down    stable    5c:07:58:79:0a:05    25G    Full    never
ifp-0/1/5              Up      Up      Up      stable    5c:07:58:79:0a:06    10G    Full    Fri Jun 05 12:37:40 GMT
+0000 2026
ifp-0/1/6              Up      Up      Up      stable    5c:07:58:79:0a:07    10G    Full    Fri Jun 05 12:37:40 GMT
+0000 2026
ifp-0/1/7              Up      Up      Up      stable    5c:07:58:79:0a:08    10G    Full    Fri Jun 05 12:37:40 GMT
+0000 2026
ifp-0/1/8              Up      Up      Up      stable    5c:07:58:79:0a:09    10G    Full    Fri Jun 05 12:37:40 GMT
+0000 2026
ifp-0/1/9              Up      Down    Down    stable    5c:07:58:79:0a:0a    25G    Full    never
ifp-0/1/10             Up      Up      Up      stable    5c:07:58:79:0a:0b    10G    Full    Fri Jun 05 12:37:40 GMT
+0000 2026
ifp-0/1/11             Up      Down    Down    stable    5c:07:58:79:0a:0c    25G    Full    never
ifp-0/1/12             Up      Up      Up      stable    5c:07:58:79:0a:0d    10G    Full    Fri Jun 05 12:37:40 GMT
+0000 2026
ifp-0/1/13             Up      Down    Down    stable    5c:07:58:79:0a:0e    25G    Full    never
ifp-0/1/14             Up      Up      Up      stable    5c:07:58:79:0a:0f    10G    Full    Fri Jun 05 12:37:40 GMT
+0000 2026
ifp-0/1/15             Up      Down    Down    stable    5c:07:58:79:0a:10    25G    Full    never
ifp-0/1/16             Up      Up      Up      stable    5c:07:58:79:0a:11    25G    Full    Fri Jun 05 12:37:40 GMT
+0000 2026
ifp-0/1/17             Up      Down    Down    stable    5c:07:58:79:0a:12    25G    Full    never
ifp-0/1/18             Up      Down    Down    stable    5c:07:58:79:0a:13    25G    Full    never
ifp-0/1/19             Up      Down    Down    stable    5c:07:58:79:0a:14    25G    Full    never
ifp-0/1/20             Up      Up      Up      stable    5c:07:58:79:0a:15    10G    Full    Fri Jun 05 12:37:40 GMT
```

```
+0000 2026
ifp-0/1/21            Up      Up      Up      stable    5c:07:58:79:0a:16    10G    Full    Fri Jun 05 12:37:40 GMT
+0000 2026
ifp-0/1/22            Up      Up      Up      stable    5c:07:58:79:0a:17    10G    Full    Fri Jun 05 12:37:40 GMT
+0000 2026
ifp-0/1/23            Up      Down    Down    stable    5c:07:58:79:0a:18    25G    Full    never
ifp-0/1/24            Up      Up      Up      stable    5c:07:58:79:0a:19    10G    Full    Fri Jun 05 12:37:40 GMT
+0000 2026
ifp-0/1/25            Up      Up      Up      stable    5c:07:58:79:0a:1a    10G    Full    Fri Jun 05 12:37:40 GMT
+0000 2026
ifp-0/1/26            Up      Up      Up      stable    5c:07:58:79:0a:1b    10G    Full    Fri Jun 05 12:37:40 GMT
+0000 2026
ifp-0/1/27            Up      Down    Down    stable    5c:07:58:79:0a:1c    25G    Full    never
ifp-0/1/28            Up      Down    Down    stable    5c:07:58:79:0a:1d    25G    Full    never
ifp-0/1/29            Up      Down    Down    stable    5c:07:58:79:0a:1e    25G    Full    never
ifp-0/1/30            Up      Down    Down    stable    5c:07:58:79:0a:1f    25G    Full    never
ifp-0/1/31            Up      Down    Down    stable    5c:07:58:79:0a:20    25G    Full    never
ifp-0/1/32            Up      Up      Up      stable    5c:07:58:79:0a:21    10G    Full    Fri Jun 05 12:37:40 GMT
+0000 2026
ifp-0/1/33            Up      Up      Up      stable    5c:07:58:79:0a:22    10G    Full    Fri Jun 05 12:37:40 GMT
+0000 2026
ifp-0/1/34            Up      Down    Down    stable    5c:07:58:79:0a:23    25G    Full    never
ifp-0/1/35            Up      Down    Down    stable    5c:07:58:79:0a:24    25G    Full    never
ifp-0/1/36            Up      Down    Down    stable    5c:07:58:79:0a:25    25G    Full    never
ifp-0/1/37            Up      Down    Down    stable    5c:07:58:79:0a:26    25G    Full    never
ifp-0/1/38            Up      Down    Down    stable    5c:07:58:79:0a:27    25G    Full    never
ifp-0/1/39            Up      Down    Down    stable    5c:07:58:79:0a:28    25G    Full    never
ifp-0/1/40            Up      Down    Down    stable    5c:07:58:79:0a:29    25G    Full    never
ifp-0/1/41            Up      Up      Up      stable    5c:07:58:79:0a:2a    10G    Full    Fri Jun 05 12:37:40 GMT
+0000 2026
ifp-0/1/42            Up      Up      Up      stable    5c:07:58:79:0a:2b    10G    Full    Fri Jun 05 12:37:40 GMT
+0000 2026
ifp-0/1/43            Up      Down    Down    stable    5c:07:58:79:0a:2c    25G    Full    never
ifp-0/1/44            Up      Down    Down    stable    5c:07:58:79:0a:2d    25G    Full    never
ifp-0/1/45            Up      Down    Down    stable    5c:07:58:79:0a:2e    25G    Full    never
ifp-0/1/46            Up      Up      Up      stable    5c:07:58:79:0a:2f    10G    Full    Fri Jun 05 12:37:40 GMT
+0000 2026
ifp-0/1/47            Up      Up      Up      stable    5c:07:58:79:0a:30    10G    Full    Fri Jun 05 12:37:40 GMT
+0000 2026
ifp-0/1/48            Up      Up      Up      stable    5c:07:58:79:0a:31    10G    Full    Fri Jun 05 12:37:40 GMT
+0000 2026
ifp-0/1/49            Up      Up      Up      stable    5c:07:58:79:0a:32    10G    Full    Fri Jun 05 12:37:40 GMT
+0000 2026
ifp-0/1/50            Up      Down    Down    stable    5c:07:58:79:0a:33    25G    Full    never
ifp-0/1/51            Up      Down    Down    stable    5c:07:58:79:0a:34    25G    Full    never
ifp-0/1/52            Up      Down    Down    stable    5c:07:58:79:0a:35    25G    Full    never
ifp-0/1/53            Up      Down    Down    stable    5c:07:58:79:0a:36    25G    Full    never
ifp-0/1/54            Up      Down    Down    stable    5c:07:58:79:0a:37    25G    Full    never
ifp-0/1/55            Up      Down    Down    stable    5c:07:58:79:0a:38    25G    Full    never
ifp-0/1/56            Up      Down    Down    stable    5c:07:58:79:0a:39    25G    Full    never
ifp-0/1/57            Up      Down    Down    stable    5c:07:58:79:0a:3a    25G    Full    never
ifp-0/1/58            Up      Down    Down    stable    5c:07:58:79:0a:3b    25G    Full    never
ifp-0/1/59            Up      Down    Down    stable    5c:07:58:79:0a:3c    25G    Full    never
ifp-0/1/60            Up      Down    Down    stable    5c:07:58:79:0a:3d    25G    Full    never
ifp-0/1/61            Up      Down    Down    stable    5c:07:58:79:0a:3e    25G    Full    never
ifp-0/1/62            Up      Down    Down    stable    5c:07:58:79:0a:3f    25G    Full    never
ifp-0/1/63            Up      Down    Down    stable    5c:07:58:79:0a:40    25G    Full    never
ifp-0/1/64            Up      Down    Down    stable    5c:07:58:79:0a:41    25G    Full    never
ifp-0/1/65            Up      Down    Down    stable    5c:07:58:79:0a:42    25G    Full    never
ifp-0/1/66            Up      Down    Down    stable    5c:07:58:79:0a:43    25G    Full    never
ifp-0/1/67            Up      Down    Down    stable    5c:07:58:79:0a:44    25G    Full    never
ifp-0/1/68            Up      Down    Down    stable    5c:07:58:79:0a:45    25G    Full    never
ifp-0/1/69            Up      Down    Down    stable    5c:07:58:79:0a:46    25G    Full    never
ifp-0/1/70            Up      Down    Down    stable    5c:07:58:79:0a:47    25G    Full    never
ifp-0/1/71            Up      Down    Down    stable    5c:07:58:79:0a:48    25G    Full    never
ifp-0/1/72            Up      Down    Down    stable    5c:07:58:79:0a:49    25G    Full    never
```

```
ifp-0/1/73            Up      Down    Down    stable    5c:07:58:79:0a:4a    25G    Full    never
ifp-0/1/74            Up      Down    Down    stable    5c:07:58:79:0a:4b    25G    Full    never
ifp-0/1/75            Up      Down    Down    stable    5c:07:58:79:0a:4c    25G    Full    never
ifp-0/1/76            Up      Down    Down    stable    5c:07:58:79:0a:4d    25G    Full    never
ifp-0/1/77            Up      Down    Down    stable    5c:07:58:79:0a:4e    25G    Full    never
ifp-0/1/78            Up      Down    Down    stable    5c:07:58:79:0a:4f    25G    Full    never
ifp-0/1/79            Up      Down    Down    stable    5c:07:58:79:0a:50    25G    Full    never
ifp-0/1/80            Up      Down    Down    stable    5c:07:58:79:0a:51    25G    Full    never
ifp-0/1/81            Up      Down    Down    stable    5c:07:58:79:0a:52    25G    Full    never
ifp-0/1/82            Up      Down    Down    stable    5c:07:58:79:0a:53    25G    Full    never
ifp-0/1/83            Up      Down    Down    stable    5c:07:58:79:0a:54    25G    Full    never
ifp-0/1/84            Up      Down    Down    stable    5c:07:58:79:0a:55    25G    Full    never
ifp-0/1/85            Up      Down    Down    stable    5c:07:58:79:0a:56    25G    Full    never
ifp-0/1/86            Up      Down    Down    stable    5c:07:58:79:0a:57    25G    Full    never
ifp-0/1/87            Up      Down    Down    stable    5c:07:58:79:0a:58    25G    Full    never
ifp-0/1/88            Up      Down    Down    stable    5c:07:58:79:0a:59    25G    Full    never
ifp-0/1/89            Up      Down    Down    stable    5c:07:58:79:0a:5a    25G    Full    never
ifp-0/1/90            Up      Down    Down    stable    5c:07:58:79:0a:5b    25G    Full    never
ifp-0/1/91            Up      Down    Down    stable    5c:07:58:79:0a:5c    25G    Full    never
ifp-0/1/92            Up      Down    Down    stable    5c:07:58:79:0a:5d    25G    Full    never
ifp-0/1/93            Up      Down    Down    stable    5c:07:58:79:0a:5e    25G    Full    never
ifp-0/1/94            Up      Down    Down    stable    5c:07:58:79:0a:5f    25G    Full    never
ifp-0/1/95            Up      Down    Down    stable    5c:07:58:79:0a:60    25G    Full    never
ifp-0/1/96            Up      Up      Up      stable    5c:07:58:79:0a:61    100G   Full    Fri Jun 05 12:37:40 GMT
+0000 2026
ifp-0/1/97            Up      Up      Up      stable    5c:07:58:79:0a:62    100G   Full    Fri Jun 05 12:37:40 GMT
+0000 2026
ifp-0/1/98            Up      Down    Down    stable    5c:07:58:79:0a:63    100G   Full    never
ifp-0/1/99            Up      Down    Down    stable    5c:07:58:79:0a:64    100G   Full    never
ifp-0/1/100           Up      Down    Down    stable    5c:07:58:79:0a:65    100G   Full    never
ifp-0/1/101           Up      Up      Up      stable    5c:07:58:79:0a:66    100G   Full    Fri Jun 05 12:37:40 GMT
+0000 2026
cpu-0/0/200           Up      Up      Up      stable    5c:07:58:79:0a:c8    100G   Full    Fri Jun 05 12:37:40 GMT
+0000 2026
cpu-0/0/201           Up      Down    Down    stable    5c:07:58:79:0a:c9    100G   Full    never
cpu-0/0/202           Up      Down    Down    stable    5c:07:58:79:0a:ca    100G   Full    never
cpu-0/0/203           Up      Down    Down    stable    5c:07:58:79:0a:cb    100G   Full    never
recycle-0/0/211       Up      Up      Up      stable    5c:07:58:79:0a:d3    100G   Full    Fri Jun 05 12:37:40 GMT
+0000 2026
recycle-0/0/212       Up      Up      Up      stable    5c:07:58:79:0a:d4    100G   Full    Fri Jun 05 12:37:40 GMT
+0000 2026
recycle-0/0/213       Up      Up      Up      stable    5c:07:58:79:0a:d5    100G   Full    Fri Jun 05 12:37:40 GMT
+0000 2026
recycle-0/0/214       Up      Up      Up      stable    5c:07:58:79:0a:d6    100G   Full    Fri Jun 05 12:37:40 GMT
+0000 2026
recycle-0/0/215       Up      Up      Up      stable    5c:07:58:79:0a:d7    100G   Full    Fri Jun 05 12:37:40 GMT
+0000 2026
recycle-0/0/216       Up      Up      Up      stable    5c:07:58:79:0a:d8    100G   Full    Fri Jun 05 12:37:40 GMT
+0000 2026
recycle-0/0/217       Up      Up      Up      stable    5c:07:58:79:0a:d9    100G   Full    Fri Jun 05 12:37:40 GMT
+0000 2026
```

### Example 4: List of All Logical Interfaces for All Instances

```
supervisor@rtbrick: op> show interface logical
Interface             Instance            Admin   Link    Oper    Outer VLAN   Inner VLAN  IPv4 Status,MTU   IPv6
Status,MTU   Type
ifl-0/1/22/0          default             Up      Up      Up      -            -           Up,1500           Up,1500
l3
ifl-0/1/22/100        default             Up      Up      Up      100          -           Up,1500           Up,1500
l3
ifl-0/1/22/200        default             Up      Up      Up      200          -           Up,1500           Up,1500
l3
ifl-0/1/22/300        default             Up      Up      Up      300          -           Up,1500           Up,1500
```

```
l3
```

Example 5: List of Logical Interfaces for an Instance

```
supervisor@rtbrick: op> show interface logical instance default
Interface              Instance            Admin   Link    Oper    Outer VLAN   Inner VLAN  IPv4 Status,MTU   IPv6
Status,MTU   Type
ifl-0/1/22/0           default             Up      Up      Up      -            -           Up,1500           Up,1500
l3
ifl-0/1/22/100         default             Up      Up      Up      100          -           Up,1500           Up,1500
l3
ifl-0/1/22/200         default             Up      Up      Up      200          -           Up,1500           Up,1500
l3
ifl-0/1/22/300         default             Up      Up      Up      300          -           Up,1500           Up,1500
l3
```

Example 6: List of All Interface Addresses

```
supervisor@rtbrick: op> show interface address
Interface              Instance            IPv4 Address             IPv4 Primary   IPv6 Address
ifl-0/1/22/0           default             192.0.2.2/30             True           fe80::5c07:58ff:fe79:a17/64
ifl-0/1/22/100         default             192.0.2.6/30             True           fe80::5c07:5800:6479:a17/64
ifl-0/1/22/200         default             192.0.2.10/30            True           fe80::5c07:5800:c879:a17/64
ifl-0/1/22/300         default             192.0.2.14/30            True           fe80::5c07:5801:2c79:a17/64
```

API To access the Operational State API that corresponds to this CLI, click here.

#### Interface Details Commands

**Syntax:**

**show interface** <option> **detail**

| Option | Description |
|---|---|
| `detail` | Without any additional option, displays detailed information for all interfaces. |
| `<interface-name> detail` | Displays detailed information for an interface. |

Example 7: Detailed Information for a Physical Interface and Its Logical Interfaces

```
supervisor@rtbrick: op> show interface ifp-0/1/22 detail
Interface: ifp-0/1/22
  Admin/Link/Operational status: Up/Up/Up
  Damping Status: stable
  Speed configured: 10G
  Speed maximum: 25G
  Duplex: Full
  Encapsulation mode: ieee
  MRU: 10000
```

```
  MTU: 9000
  FEC Bypass: True
  Maximum frame size: 10000
  Interface type: ethernet
  Interface index: 23681
  MAC: 5c:07:58:79:0a:17
  Uptime: Fri Jun 05 12:37:40 GMT +0000 2026, Holddown up/down delay: [0-10000]ms/0ms
  Description: Physical interface #22 from node 0, chip 1
  Class-of-Service: _DEFAULT_IFP_QOS_PROFILE_
  LED Status: On
  Packet statistics:
    Rx packets: 28         Tx packets: 31
    Rx bytes: 3448         Tx bytes: 3609
  Interface: ifl-0/1/22/0, Instance: default
    Admin/Link/Operational status: Up/Up/Up
    IPv4/IPv6/MPLS Status: Up/Up/Up
    IPv4/IPv6/MPLS MTU: 1500/1500/1500
    Interface type: Logical Sub interface
    Vlan Encapsulation: 802.1q
    RPF Mode: disable
    Neighbor Preference: 6
    Interface index: 22657
    MAC: 5c:07:58:79:0a:17
    IPv4 Address             IPv6 Address
    192.0.2.2/30             fe80::5c07:58ff:fe79:a17/64
    BCM statistics:
      Ingress forwarded packets: 0                  Ingress forwarded bytes: 0
      Ingress drop packets: 0                       Ingress drop bytes: 0
      Egress forwarded packets: 0                   Egress forwarded bytes: 0
      Egress drop packets: 0                        Egress drop bytes: 0
  Interface: ifl-0/1/22/100, Instance: default
    Admin/Link/Operational status: Up/Up/Up
    IPv4/IPv6/MPLS Status: Up/Up/Up
    IPv4/IPv6/MPLS MTU: 1500/1500/1500
    Interface type: Logical Sub interface
    Vlan Encapsulation: 802.1q
    Outer VLAN: 100
    RPF Mode: disable
    Neighbor Preference: 6
    Interface index: 26237057
    MAC: 5c:07:58:79:0a:17
    IPv4 Address             IPv6 Address
    192.0.2.6/30             fe80::5c07:5800:6479:a17/64
    BCM statistics:
      Ingress forwarded packets: 0                  Ingress forwarded bytes: 0
      Ingress drop packets: 0                       Ingress drop bytes: 0
      Egress forwarded packets: 0                   Egress forwarded bytes: 0
      Egress drop packets: 0                        Egress drop bytes: 0
  Interface: ifl-0/1/22/200, Instance: default
    Admin/Link/Operational status: Up/Up/Up
    IPv4/IPv6/MPLS Status: Up/Up/Up
```

```
    IPv4/IPv6/MPLS MTU: 1500/1500/1500
    Interface type: Logical Sub interface
    Vlan Encapsulation: 802.1q
    Outer VLAN: 200
    RPF Mode: disable
    Neighbor Preference: 6
    Interface index: 52451457
    MAC: 5c:07:58:79:0a:17
    IPv4 Address             IPv6 Address
    192.0.2.10/30            fe80::5c07:5800:c879:a17/64
    BCM statistics:
      Ingress forwarded packets: 0                  Ingress forwarded bytes: 0
      Ingress drop packets: 0                       Ingress drop bytes: 0
      Egress forwarded packets: 0                   Egress forwarded bytes: 0
      Egress drop packets: 0                        Egress drop bytes: 0
  Interface: ifl-0/1/22/300, Instance: default
    Admin/Link/Operational status: Up/Up/Up
    IPv4/IPv6/MPLS Status: Up/Up/Up
    IPv4/IPv6/MPLS MTU: 1500/1500/1500
    Interface type: Logical Sub interface
    Vlan Encapsulation: 802.1q
    Outer VLAN: 300
    RPF Mode: disable
    Neighbor Preference: 6
    Interface index: 78665857
    MAC: 5c:07:58:79:0a:17
    IPv4 Address             IPv6 Address
    192.0.2.14/30            fe80::5c07:5801:2c79:a17/64
    BCM statistics:
      Ingress forwarded packets: 0                  Ingress forwarded bytes: 0
      Ingress drop packets: 0                       Ingress drop bytes: 0
      Egress forwarded packets: 0                   Egress forwarded bytes: 0
      Egress drop packets: 0                        Egress drop bytes: 0
```

**MTU Profile Show Command**

**Syntax:**

**show mtu profile** <option>

| Option | Description |
|---|---|
| - | Without any additional option, displays detailed information for all MTU profiles. |
| profile-name <mtu-profile-name> | MTU Profile Name |

Example 1: Detailed Information About the MTU Profiles

```
supervisor@rtbrick: op> show mtu profile
Profile Name                          Type          Size   Action
V4_MTU_DROP                           ipv4          1472   drop
__default_pppoe__                     pppoe         1492   drop
```

Example 2: Display Information About the Specified MTU Profile

```
supervisor@rtbrick: op> show mtu profile profile-name V4_MTU_DROP
Profile Name                          Type          Size   Action
V4_MTU_DROP                           ipv4          1472   drop
```

#### Interface Statistics Commands

**Syntax:**

**show interface** <interface-name> **statistics**

| Option | Description |
|---|---|
| `statistics` | Without any additional option, displays statistics information for all interfaces. |
| `<interface-name> statistics` | Displays statistics information for an interface. |

Example 10: Statistics Information for a Physical Interface and Its Logical Interfaces

```
supervisor@rtbrick: op> show interface ifp-0/1/22 statistics
Interface:  ifp-0/1/22
  VPP statistics
      Counter              Unit      Count
      Rx                   packets   29
                           bytes     3426
      Tx                   packets   32
                           bytes     3704
      IPv4                 packets   0
      IPv6                 packets   6
      MPLS                 packets   0
      Punt                 packets   0
      Drops                packets   25
      Rx Miss              packets   0
      Rx Error             packets   0
      Rx No Buff           packets   0
      Tx Error             packets   0
  BCM statistics:
    rx_packets: 28                                        tx_packets: 31
    rx_bytes: 3448                                        tx_bytes: 3609
    ifHCInOctets: 3448                                    ifHCOutOctets: 3609
    inNonUcastPkts: 28                                    outNonUcastPkts: 31
    inErrors: 0                                           outErrors: 0
```

```
    etherStatsDropEvents: 0                                etherStatsBroadcastPkts: 0
    etherStatsUndersizePkts: 0                             etherStatsFragments: 0
    etherStatsOversizePkts: 0                              etherStatsOctets: 3448
    etherStatsPkts: 28                                     dot1dBasePortMtuExceededDiscards: 0
    outMulticastPkts: 0                                    outBroadcastPkts: 0
    bcmReceivedUndersizePkts: 0                            bcmTransmittedUndersizePkts: 0
    etherTxOversizePkts: 0                                 etherStatsJabbers: 0
    etherStatsCRCAlignErrors: 0                            dot3StatsFCSErrors: 0
    ifHCOutMulticastPkts: 31                               ifHCInMulticastPkts: 28
    ifHCOutBroadcastPckts: 0                               IfHCInBroadcastPkts: 0
    IfHCOutUcastPkts: 0                                    IfHCInUcastPkts: 0
    bcmRxFecCorrectable: 0                                 bcmRxFecUncorrectable: 0
    bcmRxRsFecBitError: 0
  Logical Interface:  ifl-0/1/22/0
     VPP statistics:
       Counter              Unit      Count
       Rx                 packets   44
                          bytes     4770
       Tx                 packets   45
                          bytes     4815
       IPv4               packets   0
       IPv6               packets   9
       MPLS               packets   0
       Punt               packets   0
       Drops              packets   39
       Rx Miss            packets   0
       Rx Error           packets   0
       Rx No Buff         packets   0
       Tx Error           packets   0
    BCM statistics:
      Ingress forwarded packets: 0                  Ingress forwarded bytes: 0
      Ingress drop packets: 0                       Ingress drop bytes: 0
      Egress forwarded packets: 0                   Egress forwarded bytes: 0
      Egress drop packets: 0                        Egress drop bytes: 0
  Logical Interface:  ifl-0/1/22/100
     VPP statistics:
       Counter              Unit      Count
       Rx                 packets   10
                          bytes     954
       Tx                 packets   10
                          bytes     940
       IPv4               packets   0
       IPv6               packets   9
       MPLS               packets   0
       Punt               packets   0
       Drops              packets   10
       Rx Miss            packets   0
       Rx Error           packets   0
       Rx No Buff         packets   0
       Tx Error           packets   0
    BCM statistics:
      Ingress forwarded packets: 0                  Ingress forwarded bytes: 0
      Ingress drop packets: 0                       Ingress drop bytes: 0
      Egress forwarded packets: 0                   Egress forwarded bytes: 0
```

```
      Egress drop packets: 0                        Egress drop bytes: 0
  Logical Interface:  ifl-0/1/22/200
     VPP statistics:
       Counter              Unit      Count
       Rx                 packets   10
                          bytes     954
       Tx                 packets   10
                          bytes     940
       IPv4               packets   0
       IPv6               packets   9
       MPLS               packets   0
       Punt               packets   0
       Drops              packets   10
       Rx Miss            packets   0
       Rx Error           packets   0
       Rx No Buff         packets   0
       Tx Error           packets   0
    BCM statistics:
      Ingress forwarded packets: 0                  Ingress forwarded bytes: 0
      Ingress drop packets: 0                       Ingress drop bytes: 0
      Egress forwarded packets: 0                   Egress forwarded bytes: 0
      Egress drop packets: 0                        Egress drop bytes: 0
  Logical Interface:  ifl-0/1/22/300
     VPP statistics:
       Counter              Unit      Count
       Rx                 packets   9
                          bytes     864
       Tx                 packets   10
                          bytes     940
       IPv4               packets   0
       IPv6               packets   8
       MPLS               packets   0
       Punt               packets   0
       Drops              packets   9
       Rx Miss            packets   0
       Rx Error           packets   0
       Rx No Buff         packets   0
       Tx Error           packets   0
    BCM statistics:
      Ingress forwarded packets: 0                  Ingress forwarded bytes: 0
      Ingress drop packets: 0                       Ingress drop bytes: 0
      Egress forwarded packets: 0                   Egress forwarded bytes: 0
      Egress drop packets: 0                        Egress drop bytes: 0
```

**Display RSFEC Error Counters**

RSFEC error counters: FEC Bit Error Counter, FEC Correctable Counter, and FEC Uncorrectable Counter (`bcmRxFecCorrectable`, `bcmRxFecUncorrectable`, and `bcmRxRsFecBitError`) are visible in the CLI.

Example for RSFEC error counter statistics

```
supervisor@rtbrick: op> show interface ifp-0/1/22 statistics
Interface:  ifp-0/1/22
```

```
  VPP statistics
      Counter              Unit      Count
      Rx                   packets   29
                           bytes     3426
      Tx                   packets   32
                           bytes     3704
      IPv4                 packets   0
      IPv6                 packets   6
      MPLS                 packets   0
      Punt                 packets   0
      Drops                packets   25
      Rx Miss              packets   0
      Rx Error             packets   0
      Rx No Buff           packets   0
      Tx Error             packets   0
  BCM statistics:
    rx_packets: 28                                          tx_packets: 31
    rx_bytes: 3448                                          tx_bytes: 3609
    ifHCInOctets: 3448                                      ifHCOutOctets: 3609
    inNonUcastPkts: 28                                      outNonUcastPkts: 31
    inErrors: 0                                             outErrors: 0
    etherStatsDropEvents: 0                                 etherStatsBroadcastPkts: 0
    etherStatsUndersizePkts: 0                              etherStatsFragments: 0
    etherStatsOversizePkts: 0                               etherStatsOctets: 3448
    etherStatsPkts: 28                                      dot1dBasePortMtuExceededDiscards: 0
    outMulticastPkts: 0                                     outBroadcastPkts: 0
    bcmReceivedUndersizePkts: 0                             bcmTransmittedUndersizePkts: 0
    etherTxOversizePkts: 0                                  etherStatsJabbers: 0
    etherStatsCRCAlignErrors: 0                             dot3StatsFCSErrors: 0
    ifHCOutMulticastPkts: 31                                ifHCInMulticastPkts: 28
    ifHCOutBroadcastPckts: 0                                IfHCInBroadcastPkts: 0
    IfHCOutUcastPkts: 0                                     IfHCInUcastPkts: 0
    bcmRxFecCorrectable: 0                                  bcmRxFecUncorrectable: 0
    bcmRxRsFecBitError: 0
  Logical Interface:  ifl-0/1/22/0
     VPP statistics:
       Counter              Unit      Count
       Rx                 packets   44
                          bytes     4770
       Tx                 packets   45
                          bytes     4815
       IPv4               packets   0
       IPv6               packets   9
       MPLS               packets   0
       Punt               packets   0
       Drops              packets   39
       Rx Miss            packets   0
       Rx Error           packets   0
       Rx No Buff         packets   0
       Tx Error           packets   0
    BCM statistics:
      Ingress forwarded packets: 0                  Ingress forwarded bytes: 0
      Ingress drop packets: 0                       Ingress drop bytes: 0
      Egress forwarded packets: 0                   Egress forwarded bytes: 0
```

```
    Egress drop packets: 0                        Egress drop bytes: 0
Logical Interface:  ifl-0/1/22/100
   VPP statistics:
     Counter              Unit      Count
     Rx                 packets   10
                        bytes     954
     Tx                 packets   10
                        bytes     940
     IPv4               packets   0
     IPv6               packets   9
     MPLS               packets   0
     Punt               packets   0
     Drops              packets   10
     Rx Miss            packets   0
     Rx Error           packets   0
     Rx No Buff         packets   0
     Tx Error           packets   0
  BCM statistics:
    Ingress forwarded packets: 0                  Ingress forwarded bytes: 0
    Ingress drop packets: 0                       Ingress drop bytes: 0
    Egress forwarded packets: 0                   Egress forwarded bytes: 0
    Egress drop packets: 0                        Egress drop bytes: 0
Logical Interface:  ifl-0/1/22/200
   VPP statistics:
     Counter              Unit      Count
     Rx                 packets   10
                        bytes     954
     Tx                 packets   10
                        bytes     940
     IPv4               packets   0
     IPv6               packets   9
     MPLS               packets   0
     Punt               packets   0
     Drops              packets   10
     Rx Miss            packets   0
     Rx Error           packets   0
     Rx No Buff         packets   0
     Tx Error           packets   0
  BCM statistics:
    Ingress forwarded packets: 0                  Ingress forwarded bytes: 0
    Ingress drop packets: 0                       Ingress drop bytes: 0
    Egress forwarded packets: 0                   Egress forwarded bytes: 0
    Egress drop packets: 0                        Egress drop bytes: 0
Logical Interface:  ifl-0/1/22/300
   VPP statistics:
     Counter              Unit      Count
     Rx                 packets   9
                        bytes     864
     Tx                 packets   10
                        bytes     940
     IPv4               packets   0
     IPv6               packets   8
     MPLS               packets   0
     Punt               packets   0
```

```
      Drops              packets   9
      Rx Miss            packets   0
      Rx Error           packets   0
      Rx No Buff         packets   0
      Tx Error           packets   0
    BCM statistics:
      Ingress forwarded packets: 0                  Ingress forwarded bytes: 0
      Ingress drop packets: 0                       Ingress drop bytes: 0
      Egress forwarded packets: 0                   Egress forwarded bytes: 0
      Egress drop packets: 0                        Egress drop bytes: 0
```

#### Show Port Map Command

The `show port map` command displays information about a physical interface's mapping with RtBrick Host port IDs and optic port IDs.

**Syntax:**

```
show port map
```

Example: Port mapping

```
supervisor@rtbrick: op> show port map
   IFP-Name            Interface-Type      Port-ID   Onlp-Port-ID   Optic-Port-ID
   cpu-0/0/0           cpu                 0         -              -
   ifp-0/1/0           ethernet            1         0              0
   ifp-0/1/1           ethernet            2         1              1
   ifp-0/1/2           ethernet            3         2              2
   ifp-0/1/3           ethernet            4         3              3
   ifp-0/1/4           ethernet            5         4              4
   ifp-0/1/5           ethernet            6         5              5
   ifp-0/1/6           ethernet            7         6              6
   ifp-0/1/7           ethernet            8         7              7
   ifp-0/1/8           ethernet            9         8              8
   ifp-0/1/9           ethernet            10        9              9
   ifp-0/1/10          ethernet            11        10             10
   ifp-0/1/11          ethernet            12        11             11
   ifp-0/1/12          ethernet            13        12             12
   ifp-0/1/13          ethernet            14        13             13
   ifp-0/1/14          ethernet            15        14             14
   ifp-0/1/15          ethernet            16        15             15
   ifp-0/1/16          ethernet            17        16             16
   ifp-0/1/17          ethernet            18        17             17
   ifp-0/1/18          ethernet            19        18             18
   ifp-0/1/19          ethernet            20        19             19
   ifp-0/1/20          ethernet            21        20             20
   ifp-0/1/21          ethernet            22        21             21
   ifp-0/1/22          ethernet            23        22             22
   ifp-0/1/23          ethernet            24        23             23
```

```
    ifp-0/1/24          ethernet            25        24              24
    ifp-0/1/25          ethernet            26        25              25
    ifp-0/1/26          ethernet            27        26              26
    ifp-0/1/27          ethernet            28        27              27
    ifp-0/1/28          ethernet            29        28              28
    ifp-0/1/29          ethernet            30        29              29
    ifp-0/1/30          ethernet            31        30              30
    ifp-0/1/31          ethernet            32        31              31
    ifp-0/1/32          ethernet            33        32              32
    ifp-0/1/33          ethernet            34        33              33
    ifp-0/1/34          ethernet            35        34              34
    ifp-0/1/35          ethernet            36        35              35
    ifp-0/1/36          ethernet            37        36              36
    ifp-0/1/37          ethernet            38        37              37
    ifp-0/1/38          ethernet            39        38              38
    ifp-0/1/39          ethernet            40        39              39
    ifp-0/1/40          ethernet            41        40              40
    ifp-0/1/41          ethernet            42        41              41
    ifp-0/1/42          ethernet            43        42              42
    ifp-0/1/43          ethernet            44        43              43
    ifp-0/1/44          ethernet            45        44              44
    ifp-0/1/45          ethernet            46        45              45
    ifp-0/1/46          ethernet            47        46              46
    ifp-0/1/47          ethernet            48        47              47
    ifp-0/1/48          ethernet            49        48              48
    ifp-0/1/49          ethernet            50        49              49
    ifp-0/1/50          ethernet            51        50              50
    ifp-0/1/51          ethernet            52        51              51
    ifp-0/1/52          ethernet            53        52              52
    ifp-0/1/53          ethernet            54        53              53
    ifp-0/1/54          ethernet            55        54              54
    ifp-0/1/55          ethernet            56        55              55
    ifp-0/1/56          ethernet            57        56              56
    ifp-0/1/57          ethernet            58        57              57
    ifp-0/1/58          ethernet            59        58              58
    ifp-0/1/59          ethernet            60        59              59
    ifp-0/1/60          ethernet            61        60              60
    ifp-0/1/61          ethernet            62        61              61
    ifp-0/1/62          ethernet            63        62              62
    ifp-0/1/63          ethernet            64        63              63
    ifp-0/1/64          ethernet            65        64              64
    ifp-0/1/65          ethernet            66        65              65
    ifp-0/1/66          ethernet            67        66              66
    ifp-0/1/67          ethernet            68        67              67
    ifp-0/1/68          ethernet            69        68              68
    ifp-0/1/69          ethernet            70        69              69
    ifp-0/1/70          ethernet            71        70              70
    ifp-0/1/71          ethernet            72        71              71
    ifp-0/1/72          ethernet            73        72              72
    ifp-0/1/73          ethernet            74        73              73
    ifp-0/1/74          ethernet            75        74              74
```

```
    ifp-0/1/75          ethernet            76        75             75
    ifp-0/1/76          ethernet            77        76             76
    ifp-0/1/77          ethernet            78        77             77
    ifp-0/1/78          ethernet            79        78             78
    ifp-0/1/79          ethernet            80        79             79
    ifp-0/1/80          ethernet            81        80             80
    ifp-0/1/81          ethernet            82        81             81
    ifp-0/1/82          ethernet            83        82             82
    ifp-0/1/83          ethernet            84        83             83
    ifp-0/1/84          ethernet            85        84             84
    ifp-0/1/85          ethernet            86        85             85
    ifp-0/1/86          ethernet            87        86             86
    ifp-0/1/87          ethernet            88        87             87
    ifp-0/1/88          ethernet            89        88             88
    ifp-0/1/89          ethernet            90        89             89
    ifp-0/1/90          ethernet            91        90             90
    ifp-0/1/91          ethernet            92        91             91
    ifp-0/1/92          ethernet            93        92             92
    ifp-0/1/93          ethernet            94        93             93
    ifp-0/1/94          ethernet            95        94             94
    ifp-0/1/95          ethernet            96        95             95
    ifp-0/1/96          ethernet            97        96             96
    ifp-0/1/97          ethernet            98        97             97
    ifp-0/1/98          ethernet            99        98             98
    ifp-0/1/99          ethernet            100       99             99
    ifp-0/1/100         ethernet            101       100            100
    ifp-0/1/101         ethernet            102       101            101
    cpu-0/0/200         cpu                 200       -              -
    cpu-0/0/201         cpu                 201       -              -
    cpu-0/0/202         cpu                 202       -              -
    cpu-0/0/203         cpu                 203       -              -
    recycle-0/0/211     recycle port        211       -              -
    recycle-0/0/212     recycle port        212       -              -
    recycle-0/0/213     recycle port        213       -              -
    recycle-0/0/214     recycle port        214       -              -
    recycle-0/0/215     recycle port        215       -              -
    recycle-0/0/216     recycle port        216       -              -
    recycle-0/0/217     recycle port        217       -              -
    sat-0/0/1           hardware internal   218       -              -
    eventor-0/0/230     hardware internal   230       -              -
    oamp-0/0/1          hardware internal   232       -              -
    olp-0/0/1           hardware internal   240       -              -
    erp-0/0/1           hardware internal   255       -              -
```

**Show Interface History Command**

The `show interface history` command displays information about the operational status of an interface.

**Syntax:**

## show interface history

Example: Interface history

```
supervisor@rtbrick: op> show interface history
Interface           Update Time                           Admin Status       Link Status        Oper Status        Speed
ifp-0/1/97          Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 100G
ifp-0/1/96          Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 100G
ifp-0/1/49          Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 10G
ifp-0/1/48          Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 10G
ifp-0/1/47          Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 10G
ifp-0/1/46          Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 10G
ifp-0/1/42          Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 10G
ifp-0/1/41          Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 10G
ifp-0/1/33          Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 10G
ifp-0/1/32          Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 10G
ifp-0/1/26          Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 10G
ifp-0/1/25          Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 10G
ifp-0/1/24          Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 10G
ifp-0/1/22          Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 10G
ifp-0/1/21          Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 10G
ifp-0/1/20          Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 10G
ifp-0/1/14          Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 10G
ifp-0/1/12          Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 10G
ifp-0/1/10          Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 10G
ifp-0/1/8           Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 10G
ifp-0/1/7           Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 10G
ifp-0/1/6           Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 10G
ifp-0/1/5           Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 10G
ifp-0/1/3           Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 10G
ifp-0/1/2           Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 10G
recycle-0/0/217     Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 100G
recycle-0/0/216     Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 100G
recycle-0/0/215     Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 100G
recycle-0/0/214     Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 100G
recycle-0/0/213     Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 100G
recycle-0/0/212     Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 100G
recycle-0/0/211     Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 100G
cpu-0/0/200         Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 100G
ifp-0/1/101         Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 100G
ifp-0/1/16          Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 25G
ifp-0/1/0           Fri Jun 05 12:37:40 GMT +0000 2026    Up                 Up                 Up                 25G
```

Example: Interface history detail

```
supervisor@rtbrick: op> show interface history detail
Interface: ifp-0/1/97
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 100G
    Forward Error Correction: rsfec
    Media Type: Copper
    Holddown: enable
Interface: ifp-0/1/96
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
```

```
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 100G
    Forward Error Correction: rsfec
    Media Type: Copper
    Holddown: enable
Interface: ifp-0/1/49
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 10G
    Media Type: Copper
    Holddown: enable
Interface: ifp-0/1/48
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 10G
    Media Type: Copper
    Holddown: enable
Interface: ifp-0/1/47
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 10G
    Media Type: Copper
    Holddown: enable
Interface: ifp-0/1/46
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 10G
    Media Type: Copper
    Holddown: enable
Interface: ifp-0/1/42
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 10G
    Media Type: Copper
    Holddown: enable
Interface: ifp-0/1/41
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 10G
    Media Type: Copper
    Holddown: enable
Interface: ifp-0/1/33
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
```

```
    Hardware Link Status: Up
    Speed: 10G
    Media Type: Fiber
    Holddown: enable
Interface: ifp-0/1/32
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 10G
    Media Type: Fiber
    Holddown: enable
Interface: ifp-0/1/26
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 10G
    Media Type: Copper
    Holddown: enable
Interface: ifp-0/1/25
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 10G
    Media Type: Copper
    Holddown: enable
Interface: ifp-0/1/24
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 10G
    Media Type: Copper
    Holddown: enable
Interface: ifp-0/1/22
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 10G
    Media Type: Copper
    Holddown: enable
Interface: ifp-0/1/21
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 10G
    Media Type: Copper
    Holddown: enable
Interface: ifp-0/1/20
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 10G
```

```
    Media Type: Copper
    Holddown: enable
Interface: ifp-0/1/14
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 10G
    Media Type: Fiber
    Holddown: enable
Interface: ifp-0/1/12
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 10G
    Media Type: Fiber
    Holddown: enable
Interface: ifp-0/1/10
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 10G
    Media Type: Fiber
    Holddown: enable
Interface: ifp-0/1/8
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 10G
    Media Type: Copper
    Holddown: enable
Interface: ifp-0/1/7
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 10G
    Media Type: Copper
    Holddown: enable
Interface: ifp-0/1/6
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 10G
    Media Type: Copper
    Holddown: enable
Interface: ifp-0/1/5
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 10G
    Media Type: Copper
    Holddown: enable
```

```
Interface: ifp-0/1/3
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 10G
    Media Type: Copper
    Holddown: enable
Interface: ifp-0/1/2
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 10G
    Media Type: Copper
    Holddown: enable
Interface: recycle-0/0/217
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 100G
    Holddown: enable
Interface: recycle-0/0/216
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 100G
    Holddown: enable
Interface: recycle-0/0/215
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 100G
    Holddown: enable
Interface: recycle-0/0/214
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 100G
    Holddown: enable
Interface: recycle-0/0/213
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 100G
    Holddown: enable
Interface: recycle-0/0/212
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 100G
    Holddown: enable
Interface: recycle-0/0/211
```

```
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 100G
    Holddown: enable
Interface: cpu-0/0/200
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 100G
    Holddown: enable
Interface: ifp-0/1/101
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 100G
    Forward Error Correction: none
    Media Type: Copper
    Holddown: enable
Interface: ifp-0/1/16
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 25G
    Media Type: Fiber
    Holddown: enable
Interface: ifp-0/1/0
  Update Time: Fri Jun 05 12:37:40 GMT +0000 2026
    Admin/Link/Operational status: Up/Up/Up
    Hardware Link Status: Up
    Speed: 25G
    Media Type: Copper
    Holddown: enable
```

### Interface Clear Commands

Clear commands allow to reset operational states.

#### Interface Statistics

This command clears interface counters.

**Syntax:**

**clear interface statistics** <option>

| Option | Description |
|---|---|
| - | Without any additional option, the command clears the counters for all interfaces. |
| <interface-name> | Clears the counters for the given interface. |

API To access the Operational State API that corresponds to this CLI, click here.

#### Clear Neighbor Interface

This command clears neighbor interface.

**Syntax:**

**clear neighbor** <interface-name>

| Option | Description |
|---|---|
| <interface-name> | Clears the given interface. |

Example:

```
supervisor@rtbrick.net: cfg> show neighbor
Instance                MAC Address          Interface          IP Address
Dynamic  Entry Time
default                 7a:0e:db:c0:00:00    ifp-0/0/0/10       11.1.1.2
true     Fri May 10 11:00:53
default                 7a:0e:db:c0:00:00    ifp-0/0/0/10       fe80::780e:db00:ac0:0
true     Fri May 10 11:00:09
ip2                     7a:0e:db:c0:00:00    ifp-0/0/0/1        10.1.1.2
true     Fri May 10 11:02:09
ip2                     7a:0e:db:c0:00:00    ifp-0/0/0/1        20.1.1.2
true     Fri May 10 11:00:53
ip2                     11:22:33:44:55:66    ifp-0/0/0/1        25.1.1.1
false    Fri May 10 11:00:02
```

Example:

```
supervisor@rtbrick.net: cfg> clear neighbor interface ifp-0/0/0/1 ip-address 10.1.1.2
Success: cleared neighbors
```

Example:

```
supervisor@rtbrick.net: cfg> show neighbor
Instance                MAC Address          Interface             IP Address
Dynamic  Entry Time
default                 7a:0e:db:c0:00:00    ifp-0/0/0/10       11.1.1.2
```

```
true     Fri May 10 11:00:53
default                 7a:0e:db:c0:00:00    ifp-0/0/0/10       fe80::780e:db00:ac0:0
true     Fri May 10 11:00:09
ip2                     7a:0e:db:c0:00:00    ifp-0/0/0/1        20.1.1.2
true     Fri May 10 11:00:53
ip2                     11:22:33:44:55:66    ifp-0/0/0/1        25.1.1.1
false    Fri May 10 11:00:02
```

#### Viewing Breakout Configuration Details

To view the breakout configuration details, enter the following command:

```
show interface <interface-name> detail
```

Example output for a specified interface.

```
supervisor@rtbrick.net: cfg> show interface ifp-0/1/0 detail
Interface:ifp-0/1/0
  Admin/Link/Operational status: Down/Down/Down
  Damping Status: stable
  Speed configured: 10G
  Speed maximum: 25G
  Duplex: Full
  Encapsulation mode: ieee
  MRU: 10000
  MTU: 9000
  Forward Error Correction: none
  Maximum frame size: 10000
  Interface type: ethernet
  Interface index: 2305
  MAC: e8:c5:7a:d5:f9:33
  Uptime:  never, Holddown up/down delay: 10000ms/0ms
  Flap count:  2
  Holddown transitions:  1
  Description: Physical interface #0 from node 0, chip 1
  LED Status: Off, LED programmed value: 10
  Packet statistics:
    Rx packets: 0          Tx packets: 714
    Rx bytes: 0            Tx bytes: 89250
```

Example of Breakout Configuration:

```
supervisor@rtbrick>SPINE01: cfg> show interface ifp-0/1/12#0 detail
Interface: ifp-0/1/12#0
  Admin/Link/Operational status: Up/Up/Up
  Damping Status: holddown
  Breakout: 4x25G
  Speed configured: 25G
```

```
  Speed maximum: 25G
  Duplex: Full
  Encapsulation mode: ieee
  MRU: 10000
  MTU: 9000
  Maximum frame size: 10000
  Interface type: ethernet
  Interface index: 100609
  MAC: e8:c5:7a:8f:56:c6
  Uptime:  Mon Nov 25 05:37:10 GMT +0000 2024, Holddown up/down delay: 10000ms/0ms
  Flap count:  1
  Holddown transitions:  1
  Description: Physical interface $$#$$12 from node 0, chip 1
  LED Status: Off, LED programmed value: 82
  Packet statistics:
    Rx packets: 3          Tx packets: 3
    Rx bytes: 381          Tx bytes: 387
```

```
supervisor@rtbrick>SPINE01: cfg> show interface ifp-0/1
/12#1 detail
Interface: ifp-0/1/12#1
  Admin/Link/Operational status: Up/Up/Up
  Damping Status: stable
  Breakout: 4x25G
  Speed configured: 25G
  Speed maximum: 25G
  Duplex: Full
  Encapsulation mode: ieee
  MRU: 10000
  MTU: 9000
  Maximum frame size: 10000
  Interface type: ethernet
  Interface index: 102657
  MAC: e8:c5:7a:8f:56:c7
  Uptime:  Mon Nov 25 05:37:10 GMT +0000 2024, Holddown up/down delay: 10000ms/0ms
  Flap count:  1
  Holddown transitions:  1
  Description: Physical interface $$#$$12 from node 0, chip 1
  LED Status: Off, LED programmed value: 4
  Packet statistics:
    Rx packets: 1          Tx packets: 1
    Rx bytes: 127          Tx bytes: 129
```

```
supervisor@rtbrick>SPINE01: cfg> show interface ifp-0/1
/12#2 detail
Interface: ifp-0/1/12#2
  Admin/Link/Operational status: Up/Up/Up
  Damping Status: stable
```

```
  Breakout: 4x25G
  Speed configured: 25G
  Speed maximum: 25G
  Duplex: Full
  Encapsulation mode: ieee
  MRU: 10000
  MTU: 9000
  Maximum frame size: 10000
  Interface type: ethernet
  Interface index: 104705
  MAC: e8:c5:7a:8f:56:c8
  Uptime:  Mon Nov 25 05:37:10 GMT +0000 2024, Holddown up/down delay: 10000ms/0ms
  Flap count:  1
  Holddown transitions:  1
  Description: Physical interface $$#$$12 from node 0, chip 1
  LED Status: Off, LED programmed value: 0
  Packet statistics:
    Rx packets: 1          Tx packets: 1
    Rx bytes: 127          Tx bytes: 129
```

```
supervisor@rtbrick>SPINE01: cfg> show interface ifp-0/1
/12#3 detail
Interface: ifp-0/1/12#3
  Admin/Link/Operational status: Up/Up/Up
  Damping Status: stable
  Breakout: 4x25G
  Speed configured: 25G
  Speed maximum: 25G
  Duplex: Full
  Encapsulation mode: ieee
  MRU: 10000
  MTU: 9000
  Maximum frame size: 10000
  Interface type: ethernet
  Interface index: 106753
  MAC: e8:c5:7a:8f:56:c9
  Uptime:  Mon Nov 25 05:37:10 GMT +0000 2024, Holddown up/down delay: 10000ms/0ms
  Flap count:  1
  Holddown transitions:  1
  Description: Physical interface $$#$$12 from node 0, chip 1
  LED Status: Off, LED programmed value: 0
  Packet statistics:
    Rx packets: 1          Tx packets: 1
    Rx bytes: 127          Tx bytes: 129
```

## 6.2. ARP/ND

### 6.2.1. ARP/ND Overview

RBFS allows you to set the timer information for ARP and ND routes. The timer information specifies how frequently a device sends messages to its neighbor. RBFS provides a logical interface with which you can configure timers for neighbor routers. These timers are essential for protocols such as ARP and ND.

RBFS supports timers for the following attributes:

**Gratuitous ARP Interval**

The Gratuitous ARP is sent as a broadcast by a node to communicate its IP address to MAC address mapping on the network. The GARP timer enables you to specify the interval time based on which GARP can be communicated.

**Neighbor Probe Interval**

RBFS allows you to configure the neighbor probe interval for the specified interface. This attribute is used to ensure that the neighbor is available or not.

**Router Advertisement Interval**

RBFS allows you to configure router advertisement interval. Router advertisement includes route information to show the network hosts that the router is operational. The router sends these messages periodically within a time range specified with minimum and maximum values. The Router Advertisement Interval timer applies only to IPv6.

**Neighbor Scan Interval**

It specifies the time interval for neighbor router scanning. It scans the ARP table of a neighbor router to determine which IP addresses are active.

**ARP Throttle Interval**

ARP throttling is a method of rate limiting of ARP packets and it safeguards the router by limiting too many ARP requests triggered by incoming traffic. RBFS allows you to configure the time interval for ARP throttling.

```
**Support for Permanent ARP Entry on IFLs**
```

RBFS supports the Permanent ARP Entry on IFLs functionality. This functionality allows devices to continually attempt address resolution for IPv4 or Neighbor Discovery for IPv6 on critical IP addresses within a LAN. The device can dynamically resolve the MAC addresses associated with these specified IPs, even if the GARP is missed from the neighbor.

**Supported Platforms**

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

### 6.2.2. ARP/ND Configuration

### Configuration Hierarchy

The diagram illustrates the Neighbor Timer configuration hierarchy.

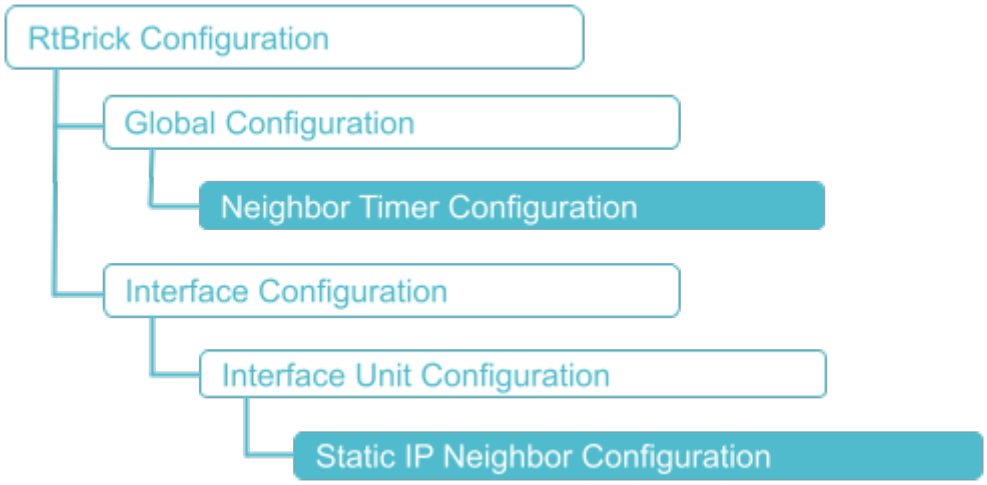



*Figure 115. Neighbor Timer Configuration Hierarchy*

### Configuration Syntax and Commands

#### Neighbor Timer Configuration

The following sections describe the interface configuration syntax and commands.

**Syntax:**

**set global neighbor** <attribute> <value>

| Attribute | Description |
|---|---|
| `garp-interval` | Gratuitous ARP interval. The value is in seconds.<br>Range: 1 to 1000 seconds<br>Default: 60 |
| `probe-interval` | Neighbor probe interval. The value is in seconds.<br>Range: 1 to 1000 seconds<br>Default: 10 |
| `ra-interval` | Router advertisement interval. The value is in seconds.<br>Range: 1 to 1000 seconds<br>Default: 10 |
| `scan interval` | Neighbor scan interval. The value is in seconds.<br>Range: 1 to 1000 seconds<br>Default: 60 |
| `throttle-interval` | ARP throttle interval. The value is in seconds.<br>Range: 1 to 1000 seconds<br>Default: 10 |

Example: Neighbor timer Configuration

```
{
    "rtbrick-config:neighbor": {
      "garp-interval": 10,
      "probe-interval": 120,
      "scan-interval": 120,
      "throttle-interval": 120,
      "ra-interval": 120
    }
  }
```

#### Static IP Neighbor Configuration

This section describes configuration options at static IP neighbors.

**Syntax:**

**set interface** <interface-name> **unit** <unit-id> **neighbor** <attribute> <value>

| Attribute | Description |
|---|---|
| `<interface-name>` | Name of the interface. Examples: ifp-0/0/1. |
| `<unit-id>` | Create a logical interface (also referred to as a sub-interface) under the physical interface. |
| `IPv4/IPv6 <ip-address>` | Neighbor IPv4 or IPv6 address. |
| `MAC <mac-address>` | Neighbor MAC address. |

Example: Static IP Neighbor Configuration

```
supervisor@rtbrick>LEAF01: cfg> show config
{
  "data": {
    "rtbrick-config:interface": [
      {
        "name": "ifp-0/1/5",
        "unit": [
          {
            "unit-id": 1,
            "neighbor": {
              "ipv4": [
                {
                  "address4": "198.51.100.10",
                  "mac": "11:11:11:11:11:11"
                }
              ]
            }
          }
```

```
        ]
      }
    ]
  }
}
```

#### Disabling Gratuitous ARP on Interface

You can disable the sending of Gratuitous ARP (GARP) on an interface, which is enabled by default on all interfaces.

> To enable GARP on an interface, you need to delete the `garp-disable true` configuration.

**Syntax:**

**set interface** <interface-name> **unit** <unit-id> **garp-disable true**

| Attribute | Description |
|---|---|
| interface <interface-name> | Name of the interface. Examples: ifp-0/0/1. |
| garp-disable true | Disables Gratuitous ARP (GARP). Default is GARP enabled. |

Example: Disabling Gratuitous ARP

```
supervisor@rtbrick.net: cfg> set interface ifp-0/0/0 unit 100 garp-disable true
```

#### Enabling Permanent ARP Entry on IFLs

You can enable the Permanent ARP Entry on IFLs functionality.

**Syntax:**

**set interface** <interface> **unit** <unit id> **neighbor** < **ipv4** | **ipv6** > <ip address> **dynamic true**

| Attribute | Description |
|---|---|
| `<interface-name>` | Name of the interface. Example: ifp-0/0/1. |
| `<unit-id>` | Create a logical interface (also referred to as a sub-interface) under the physical interface. |
| `neighbor IPv4/IPv6 <ip-address>` | Neighbor IPv4 or IPv6 address. |
| `dynamic` | Dynamic resolution mode. |

Example: Enable Permanent ARP Entry on IFLs

```
{
  "ietf-restconf:data": {
    "rtbrick-config:interface": [
      {
        "name": "ifp-0/0/0",
        "host-if": "S1-1-SN",
        "unit": [
          {
            "unit-id": 10,
            "neighbor": {
              "ipv4": [
                {
                  "address4": "10.2.2.2",
                  "dynamic": "true"
                }
              ]
            }
          }
        ]
      },
      {
        "name": "ifp-0/0/1",
        "host-if": "S1-2-SN"
      },
      {
        "name": "ifp-0/0/2",
        "host-if": "S1-3-L1"
      },
      {
        "name": "ifp-0/0/3",
        "host-if": "S1-4-L1"
      }
    ]
  }
}
```

#### Host-Path Rate Limiting for Unresolved ARP/ND Traffic

RBFS supports host-path traffic policing for packets generated due to ARP/ND resolution failures for directly connected destinations. When ARP/ND resolution fails, traffic destined for unresolved next hops can be redirected to the control plane, causing excessive host-path load. By default, this feature is enabled with a default policer value of 10Mbps.

**Syntax**:

**set forwarding-options hostpath policer** <policer>

| Option | Description |
| --- | --- |
| `<policer>` | Name of the policer to be applied to the host path. |

The following configuration example attaches a specific policer named 'policer-hostpath' to the `hostpath` to ensure that management and protocol traffic does not overwhelm the processor during traffic spikes.

```
set forwarding-options hostpath policer policer-hostpath
```

```
{
  "forwarding-options": {
    "hostpath": {
      "policer": {
        "policer-name": policer-hostpath
      }
    }
  }
}
```

#### Configuring Route Selection for Unnumbered Interfaces

You can configure the preference of directly connected (ARP/ND) routes. Use the following CLI command to configure the route preference for ARP/ND routes.

**Syntax:**

**set interface** <interface-name> **unit** <unit-id> **neighbor-preference** <value>

| Option | Description |
| --- | --- |
| `interface` | Name of the interface. |
| `unit` | Specifies the logical sub-interface partition. |
| `neighbor-preference <value>` | The specific routing or protocol attribute for the preferred neighbor. Allowed range: 0 - 300. |

Example:

The configuration maps the neighbor-preference attribute value of 100 for interface ifp-0/0/1 and its logical partition unit 0.

```
set interface ifp-0/0/1 unit 0 neighbor-preference 100
```

```
{
  "interface": {
```

```
    "name": "ifp-0/0/1",
    "unit": [
      {
        "id": 0,
        "neighbor-preference": 100
      }
    ]
  }
}
```

### 6.2.3. ARP/ND Operational Commands

#### Show Commands

##### Neighbor Timer Show Commands

**Syntax:**

**show neighbor** <option>

| Option | Description |
|---|---|
| - | Without any option, the commands display the information for all neighbors. |
| instance <instance_name> | Displays summary of the specified neighbor instance |
| IPv4 | Displays neighbor summary for the specified IPv4 address. |
| IPv6 | Displays neighbor summary for the specified IPv6 address. |
| request <...> | Displays information about GARP requests made. |

Example 1: Summary of neighbor

```
supervisor@rtbrick: op> show neighbor
Instance                MAC Address          Interface             IP Address                                  Dynamic
Entry Time
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/0          192.0.2.1                                   true
Fri Jun 05 12:42:09
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/100        192.0.2.5                                   true
Fri Jun 05 12:42:09
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/200        192.0.2.9                                   true
Fri Jun 05 12:42:09
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/300        192.0.2.13                                  true
Fri Jun 05 12:42:09
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/0          fe80::e8c5:7aff:fe8f:565f                   true
Fri Jun 05 12:41:10
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/100        fe80::e8c5:7a00:648f:565f                   true
Fri Jun 05 12:41:10
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/200        fe80::e8c5:7a00:c88f:565f                   true
Fri Jun 05 12:41:10
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/300        fe80::e8c5:7a01:2c8f:565f                   true
Fri Jun 05 12:41:10
```

Example 2: Summary of neighbor instance

```
supervisor@rtbrick: op> show neighbor instance default
Instance                MAC Address          Interface             IP Address                                  Dynamic
Entry Time
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/0          192.0.2.1                                   true
Fri Jun 05 12:42:09
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/100        192.0.2.5                                   true
Fri Jun 05 12:42:09
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/200        192.0.2.9                                   true
Fri Jun 05 12:42:09
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/300        192.0.2.13                                  true
Fri Jun 05 12:42:09
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/0          fe80::e8c5:7aff:fe8f:565f                   true
Fri Jun 05 12:41:10
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/100        fe80::e8c5:7a00:648f:565f                   true
Fri Jun 05 12:41:10
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/200        fe80::e8c5:7a00:c88f:565f                   true
Fri Jun 05 12:41:10
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/300        fe80::e8c5:7a01:2c8f:565f                   true
Fri Jun 05 12:41:10
```

Example 3: Summary of the neighbor for the specified address family

```
supervisor@rtbrick: op> show neighbor ipv4
Instance                MAC Address          Interface             IP Address                                  Dynamic
Entry Time
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/0          192.0.2.1                                   true
Fri Jun 05 12:42:09
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/100        192.0.2.5                                   true
Fri Jun 05 12:42:09
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/200        192.0.2.9                                   true
Fri Jun 05 12:42:09
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/300        192.0.2.13                                  true
Fri Jun 05 12:42:09
```

```
supervisor@rtbrick: op> show neighbor ipv6
Instance                MAC Address          Interface             IP Address                                  Dynamic
Entry Time
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/0          fe80::e8c5:7aff:fe8f:565f                   true
Fri Jun 05 12:41:10
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/100        fe80::e8c5:7a00:648f:565f                   true
Fri Jun 05 12:41:10
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/200        fe80::e8c5:7a00:c88f:565f                   true
Fri Jun 05 12:41:10
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/300        fe80::e8c5:7a01:2c8f:565f                   true
Fri Jun 05 12:41:10
```

Example 4: Summary of the static IP neighbor

```
supervisor@rtbrick: op> show neighbor
Instance                MAC Address          Interface             IP Address                                  Dynamic
Entry Time
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/0          192.0.2.1                                   true
Fri Jun 05 12:42:09
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/100        192.0.2.5                                   true
Fri Jun 05 12:42:09
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/200        192.0.2.9                                   true
Fri Jun 05 12:42:09
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/300        192.0.2.13                                  true
```

```
Fri Jun 05 12:42:09
default                e8:c5:7a:8f:56:5f    ifl-0/1/22/0          fe80::e8c5:7aff:fe8f:565f                  true
Fri Jun 05 12:41:10
default                e8:c5:7a:8f:56:5f    ifl-0/1/22/100        fe80::e8c5:7a00:648f:565f                  true
Fri Jun 05 12:41:10
default                e8:c5:7a:8f:56:5f    ifl-0/1/22/200        fe80::e8c5:7a00:c88f:565f                  true
Fri Jun 05 12:41:10
default                e8:c5:7a:8f:56:5f    ifl-0/1/22/300        fe80::e8c5:7a01:2c8f:565f                  true
Fri Jun 05 12:41:10
```

#### Neighbor Request

This command is used to display detailed information about various ARP neighbor requests.

**Syntax:**

**show neighbor request** <option>

| Option | Description |
|---|---|
| - | Without any option, the commands display the information about all ARP requests made. |
| type | This option allows you to specify the type of neighbor request. |
| arp-proxy | Displays information about ARP proxy requests. |
| arp-reply | Shows ARP reply messages. |
| arp-request | Displays ARP request messages. |
| arp-static | Shows static ARP entries. |
| garp | Displays Gratuitous ARP (GARP) messages. |

Example 1: Summary of the request

```
supervisor@rtbrick: op> show neighbor request
Interface            IP Address                                  Type              Entry Time
ifl-0/1/22/0         192.0.2.2                                   garp              Fri Jun 05 12:41:06
ifl-0/1/22/100       192.0.2.6                                   garp              Fri Jun 05 12:41:06
ifl-0/1/22/200       192.0.2.10                                  garp              Fri Jun 05 12:41:06
ifl-0/1/22/300       192.0.2.14                                  garp              Fri Jun 05 12:41:06
ifl-0/1/22/0         fe80::5c07:58ff:fe79:a17                    garp              Fri Jun 05 12:41:06
ifl-0/1/22/100       fe80::5c07:5800:6479:a17                    garp              Fri Jun 05 12:41:06
ifl-0/1/22/200       fe80::5c07:5800:c879:a17                    garp              Fri Jun 05 12:41:06
ifl-0/1/22/300       fe80::5c07:5801:2c79:a17                    garp              Fri Jun 05 12:41:06
```

Example 2: Summary of the request type GARP

```
supervisor@rtbrick: op> show neighbor request type garp
Interface            IP Address                                  Type              Entry Time
ifl-0/1/22/0         192.0.2.2                                   garp              Fri Jun 05 12:41:06
ifl-0/1/22/100       192.0.2.6                                   garp              Fri Jun 05 12:41:06
ifl-0/1/22/200       192.0.2.10                                  garp              Fri Jun 05 12:41:06
ifl-0/1/22/300       192.0.2.14                                  garp              Fri Jun 05 12:41:06
```

```
ifl-0/1/22/0         fe80::5c07:58ff:fe79:a17                    garp              Fri Jun 05 12:41:06
ifl-0/1/22/100       fe80::5c07:5800:6479:a17                    garp              Fri Jun 05 12:41:06
ifl-0/1/22/200       fe80::5c07:5800:c879:a17                    garp              Fri Jun 05 12:41:06
ifl-0/1/22/300       fe80::5c07:5801:2c79:a17                    garp              Fri Jun 05 12:41:06
```

#### Neighbor History

This command is used to display information for deleted or disconnected neighbors. It allows you to retrieve information for the latest 1,000 deleted neighbor entries.

**Syntax:**

**show neighbor history** <option>

| Option | Description |
|---|---|
| - | Without any option, the command displays the history of all neighbors. |
| instance | Displays the neighbor's history for the specified instance. |
| interface | Displays history of a specified interface. |
| ipv4 | Displays history of a specified IPv4 neighbor. |
| ipv6 | Displays history of a specified IPv6 neighbor. |

Example 1: Neighbor history

```
supervisor@rtbrick: op> show neighbor history
Instance                MAC Address          Interface             IP Address                                  Dynamic
Create Time          Delete Time
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/0          192.0.2.1                                   true
Fri Jun 05 12:39:53  Fri Jun 05 12:40:10
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/0          fe80::e8c5:7aff:fe8f:565f                   true
Fri Jun 05 12:38:54  Fri Jun 05 12:40:10
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/100        192.0.2.5                                   true
Fri Jun 05 12:39:53  Fri Jun 05 12:40:10
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/100        fe80::e8c5:7a00:648f:565f                   true
Fri Jun 05 12:38:54  Fri Jun 05 12:40:10
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/200        192.0.2.9                                   true
Fri Jun 05 12:39:53  Fri Jun 05 12:40:10
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/200        fe80::e8c5:7a00:c88f:565f                   true
Fri Jun 05 12:38:54  Fri Jun 05 12:40:10
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/300        192.0.2.13                                  true
Fri Jun 05 12:39:53  Fri Jun 05 12:40:10
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/300        fe80::e8c5:7a01:2c8f:565f                   true
Fri Jun 05 12:38:54  Fri Jun 05 12:40:10
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/0          192.0.2.1                                   true
Fri Jun 05 12:40:41  Fri Jun 05 12:40:49
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/0          fe80::e8c5:7aff:fe8f:565f                   true
Fri Jun 05 12:40:32  Fri Jun 05 12:40:49
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/100        fe80::e8c5:7a00:648f:565f                   true
Fri Jun 05 12:40:32  Fri Jun 05 12:40:49
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/200        fe80::e8c5:7a00:c88f:565f                   true
Fri Jun 05 12:40:32  Fri Jun 05 12:40:49
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/300        fe80::e8c5:7a01:2c8f:565f                   true
Fri Jun 05 12:40:32  Fri Jun 05 12:40:49
default                 e8:c5:7a:8f:56:5f    ifl-0/1/22/0          192.0.2.1                                   true
Fri Jun 05 12:42:09  Fri Jun 05 12:42:17
```

```
default                e8:c5:7a:8f:56:5f    ifl-0/1/22/0          fe80::e8c5:7aff:fe8f:565f                  true
Fri Jun 05 12:41:10  Fri Jun 05 12:42:17
default                e8:c5:7a:8f:56:5f    ifl-0/1/22/100        192.0.2.5                                  true
Fri Jun 05 12:42:09  Fri Jun 05 12:42:18
default                e8:c5:7a:8f:56:5f    ifl-0/1/22/100        fe80::e8c5:7a00:648f:565f                  true
Fri Jun 05 12:41:10  Fri Jun 05 12:42:18
default                e8:c5:7a:8f:56:5f    ifl-0/1/22/200        192.0.2.9                                  true
Fri Jun 05 12:42:09  Fri Jun 05 12:42:18
default                e8:c5:7a:8f:56:5f    ifl-0/1/22/200        fe80::e8c5:7a00:c88f:565f                  true
Fri Jun 05 12:41:10  Fri Jun 05 12:42:18
```

Example 2: History of a specified nieghbor interface

```
supervisor@rtbrick: op> show neighbor history interface ifl-0/1/22/0
Instance               MAC Address          Interface             IP Address                                 Dynamic
Create Time          Delete Time
default                e8:c5:7a:8f:56:5f    ifl-0/1/22/0          192.0.2.1                                  true
Fri Jun 05 12:39:53  Fri Jun 05 12:40:10
default                e8:c5:7a:8f:56:5f    ifl-0/1/22/0          fe80::e8c5:7aff:fe8f:565f                  true
Fri Jun 05 12:38:54  Fri Jun 05 12:40:10
default                e8:c5:7a:8f:56:5f    ifl-0/1/22/0          192.0.2.1                                  true
Fri Jun 05 12:40:41  Fri Jun 05 12:40:49
default                e8:c5:7a:8f:56:5f    ifl-0/1/22/0          fe80::e8c5:7aff:fe8f:565f                  true
Fri Jun 05 12:40:32  Fri Jun 05 12:40:49
default                e8:c5:7a:8f:56:5f    ifl-0/1/22/0          192.0.2.1                                  true
Fri Jun 05 12:42:09  Fri Jun 05 12:42:17
default                e8:c5:7a:8f:56:5f    ifl-0/1/22/0          fe80::e8c5:7aff:fe8f:565f                  true
Fri Jun 05 12:41:10  Fri Jun 05 12:42:17
```

# 6.3. ACLs

## 6.3.1. ACL Overview

### ACL Use Cases

In RBFS, Access Control Lists (ACL) serve multiple purposes:

- Provide security by traffic filtering. This applies to both host and transit traffic. ACLs for traffic filtering are user-defined by configuration.
- Redirecting control traffic to the CPU. Such protocol ACLs also referred to as trap rules, are automatically created by the respective protocol, and do not need to be configured.
- Classifying traffic for differentiated QoS treatment. This is a special form of ACL referred to as a multi-field (MF) classifier. For more information about MF classifiers, please refer to the HQoS Configuration Guide.

### ACL Components and Processing

User-defined ACLs consist of rules and ordinals. In case of multiple matching ACL rules, you can use priorities to define the result of the ACL.

- Rules - A rule is a named ACL entry that typically contains one or multiple match criteria and an action.

- Ordinals - An ordinal is solely a numbered configuration object. A rule can consist of multiple ordinals. Ordinals help to structure the configuration. In RBFS, it makes no difference if you configure one rule with multiple ordinals or multiple rules with one ordinal each. Please note ordinals do not define the order of processing.
- Scope - ACLs generally apply globally. In particular, they are not applied to interfaces. You can, however configure an interface as a match criteria.
- Priorities - ACL entry priorities are used to define the processing of multiple matching ACL rules. In RBFS, by default, all ACL entries have the same priority, and there is no specific order. For example, if one ACL rule shall permit ICMP traffic from a specific prefix, and another rule shall deny any other ICMP traffic, it will by default result in a conflict as an ICMP packet matches both rules. To ensure that the more specific rule matches first, you can set its priority to higher. When the ACL priority value is set to a lower number, priority is higher.

**Prefix Lists**

A prefix list is a named list of prefixes. Instead of listing multiple individual prefixes in a match rule of the ACL itself, you can reference a list that contains the prefixes, and thereby apply a common action to all matching prefixes. This helps to maintain lists and reuse them in multiple ACL rules.

Prefix lists can be used in ACL for permitting/denying traffic and in Multifield Classifier (MFC) for classifying traffic. This guide describes how to configure prefix lists and apply them in user-defined ACLs as firewall filters and apply prefix lists in MFC for traffic classification. For more information about applying prefix lists to MF classifiers, please refer to the *HQoS configuration Guide*.

When a prefix list is configured and referenced in an ACL, it is internally first added to an intermediate ACL configuration table. For each prefix, one separate rule is added to the final ACL configuration table. This is different from a prefix match in the ACL rule itself that is directly added to the ACL configuration table. A dedicated range of ordinals (200001-4294967295) is reserved to expand ACL rules when using prefix lists. If configured, the priority will be copied from the prefix list ACL configuration to all the expanded ACL rules.

When using prefix lists, the following restrictions apply:

- You cannot configure the same prefix-list name to match the source prefix list and destination prefix list.
- You cannot configure both the source prefix and source prefix list on the same ACL configuration.
- You cannot configure both the destination prefix and destination prefix list on the same ACL configuration.

**Guidelines and Limitations**

- ACL and mirror configurations are not supported in virtual deployments.

**Supported Platforms**

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

## 6.3.2. ACL Configuration

For configuring an access control list, you define filter criteria (with match conditions) for the packets and an action for the device to take if the packets match the filtering criteria.

### Configuration Hierarchy

The diagram illustrates the ACL configuration hierarchy.

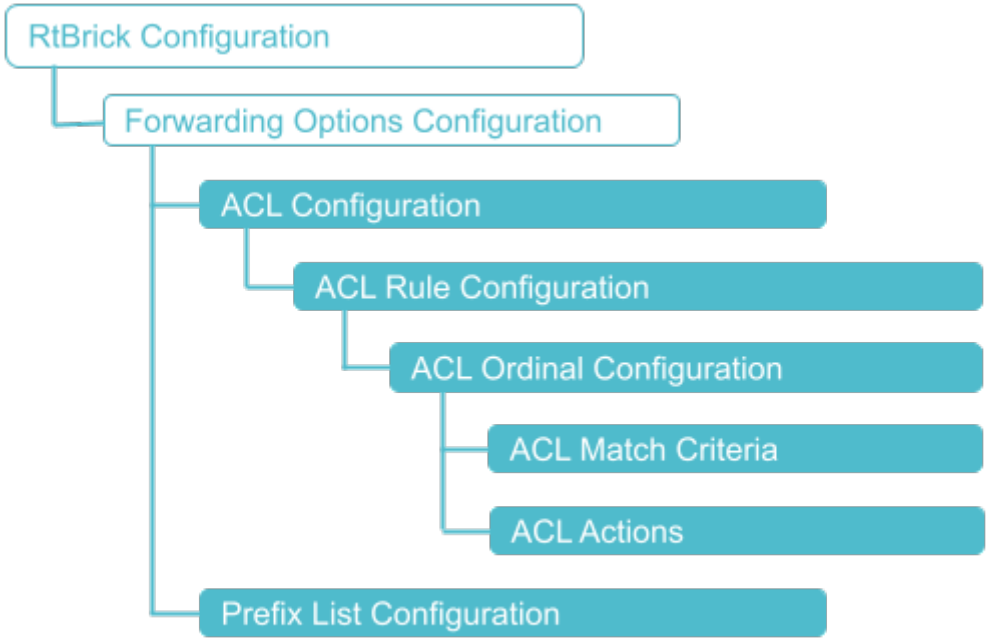



*Figure 116. ACL Configuration Hierarchy*

### Configuration Syntax and Commands

The following sections describe the ACL configuration syntax and commands.

#### Configuring ACLs

**Syntax:**

```
set forwarding-options acl [l2 | l3v4 | l3v6] rule <name> ordinal <number> <option>
<attribute> <value>
```

| Options | Description |
|---|---|
| `<name>` | Name of the ACL rule. |
| `<number>` | Specifies the ordinal number. |
| `match <…>` | Match configuration hierarchy. Refer to the Configuring ACL Match Criteria section for the ACL match criteria configuration. |

| Options | Description |
| --- | --- |
| `action <…>` | Action configuration hierarchy. Refer to the Configuring ACL Actions section for the ACL actions configuration. |
| `priority <priority>` | Specifies the ACL priority value. The default entry priority for user-defined ACLs changes to 500. The configurable ACL entry priority range is 100 to 20,000. A lower number indicates higher priority.<br>The user-defined ACL priority range is 100 to 20,000, while BGP FlowSpec priority ranges from 1,501 to 2,500, which prevents conflicts during configuration. |

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Configuring ACL Match Criteria

The L3V6 ACL with match criteria 'logical interface' is not supported, but the same is supported for L3V4 ACL.

**set forwarding-options acl** [ **l2** | **l3v4** | **l3v6** ] **rule** <rulename> **ordinal** <ordinal_value> **match** <attribute> <value>

| Attribute | Description |
| --- | --- |
| `destination-mac <destination-mac>` | ACL L2 destination mac match. |
| `destination-ipv4-prefix <destination-ipv4-prefix>` | ACL L3 IPv4 destination prefix match. |
| `destination-ipv4-prefix-list <destination-ipv4-prefix-list>` | ACL destination IPv4 prefix-list name. You can apply a prefix list that is previously configured. Refer to section Configuring Prefix Lists. |
| `destination-l4-port <destination-l4-port>` | ACL L4 destination port match. |
| `destination-ipv4-local true` | Indicates whether match support is enabled for all traffic destined for the routers' IP addresses. To disable the configuration, use the `delete` form of the command. |
| `destination-ipv6-prefix <destination-ipv6-prefix>` | ACL L3 IPv6 destination prefix match. |
| `destination-ipv6-prefix-list <destination-ipv6-prefix-list>` | ACL destination IPv6 prefix-list name. You can apply a prefix list that is previously configured. Refer to section Configuring Prefix Lists. |
| `destination-ipv6-local true` | Indicates whether match support is enabled for all traffic destined for the routers' IP addresses. To disable the configuration, use the `delete` form of the command. |

| Attribute | Description |
| --- | --- |
| `direction ingress` | ACL L2/L3 direction match. Currently, only the ingress direction is supported. |
| `ethertype <ethertype>` | ACL L2 EtherType match. |
| `inner-tag-protocol-id <inner-tag-protocol-id>` | ACL L2 inner TPID match. |
| `inner-vlan <inner-vlan>` | ACL L2 inner-VLAN match. |
| `inner-vlan-cfi <inner-vlan-cfi>` | ACL L2 inner-VLAN CFI match. |
| `inner-vlan-priority <inner-vlan-priority>` | ACL L2 inner-VLAN priority match. |
| `interface <interface>` | Interface match. |
| `ip-options true` | Match if the IPv4 packet has options. Supported value: true. |
| `ip-protocol <protocol>` | ACL IP protocol value match such as TCP, UDP, ICMP. |
| `ipv4-dscp <ipv4-dscp>` | IPv4 DSCP value. |
| `ipv4-tos <ipv4-tos>` | IPv4 ToS value. |
| `ipv6-tc <ipv6-tc>` | Codepoint class value. |
| `logical-interface <logical-interface>` | Logical interface match. This is supported only for 'l3v4' ACL. |
| `source-ipv4-prefix <source-ipv4-prefix>` | ACL L3 IPv4 source prefix match. |
| `source-ipv4-prefix-list <source-ipv4-prefix-list>` | ACL source IPv4 prefix-list name. You can apply a prefix list that is previously configured. Refer to section Configuring Prefix Lists. |
| `source-ipv6-prefix <source-ipv6-prefix>` | Configure ACL L3 IPv6 source prefix match. |
| `source-ipv6-prefix-list <source-ipv6-prefix-list>` | ACL source IPv6 prefix-list name. You can apply a prefix list that is previously configured. Refer to section Configuring Prefix Lists. |
| `source-l4-port <source-l4-port>` | ACL L4 source port match. |
| `outer-tag-protocol-id <outer-tag-protocol-id>` | ACL L2 outer TPID match. |
| `outer-vlan <outer-vlan>` | ACL L2 outer-VLAN match. |
| `outer-vlan-cfi <outer-vlan-cfi>` | ACL L2 outer VLAN CFI match. |
| `outer-vlan-priority <outer-vlan-priority>` | ACL L2 outer VLAN priority match. |
| `source-mac <source-mac>` | ACL L2 source MAC match. |
| `traffic-class <class>` | Forward class value. Supported values: class-0 to class-7, class-all. |
| `ttl <ttl>` | IPv4 time-to-live value. |
| `match-mpls-traffic true` | Match single MPLS label termination. To disable, use the `delete` form of the command. |

For information on supported ACL match criteria and actions for user-defined, Layer-2, and multi-field classifier ACLs, see the ACL Match/Action Matrix section of the Platform Guide.

API To access the RESTCONF API that corresponds to this CLI, click here.

Example 1: Layer 2 Match Configuration

```
{
    "rtbrick-config:acl": {
       "l2": {
          "rule": [
            {
              "rule-name": "a10nsp-drop-lag-2",
              "ordinal": [
                {
                  "ordinal-value": 1,
                  "match": {
                    "direction": "ingress",
                    "interface": "lag-2",
                    "outer-vlan-priority": 1
                  },
                  "action": {
                    "drop": "true",
                    "statistics": "true"
                  }
                },
```

Example 2: Layer 3 IPv4 Match Configuration

```
{
    "rtbrick-config:acl": {
      "l3v4": {
        "rule": [
          {
            "rule-name": "rtb_firewall_two",
            "ordinal": [
              {
                "ordinal-value": 1000,
                "match": {
                  "direction": "ingress",
                  "source-ipv4-prefix": "198.51.100.50/24",
                  "source-l4-port": 8080
                },
                "action": {
                  "drop": "true"
                }
              }
            ]
          },
```

```
          {
            "rule-name": "rule2",
            "ordinal": [
              {
                "ordinal-value": 5,
                "match": {
                  "direction": "ingress",
                  "interface": "ifp-0/0/1"
                }
              }
            ]
          }
        ]
      }
    }
  }
```

Example 3: Layer 3 IPv6 Match Configuration

```
{
    "rtbrick-config:l3v6": {
      "rule": [
        {
          "rule-name": "rtb_firewall_two",
          "ordinal": [
            {
              "ordinal-value": 1000,
              "match": {
                "direction": "ingress",
                "source-ipv6-prefix": "2001:db8:0:11::/32",
                "source-l4-port": 8080
              },
              "action": {
                "permit": "true"
              }
            }
          ]
        }
      ]
    }
  }
```

Example 4: Match support for all traffic destined for any of the router’s IP addresses

```
{
    "rtbrick-config:acl": {
      "l3v4": {
        "rule": [
          {
```

```
            "rule-name": "rule4",
            "ordinal": [
              {
                "ordinal-value": 4,
                "match": {
                  "direction": "ingress",
                  "destination-ipv4-local": "true"
                },
                "action": {
                  "drop": "true"
                }
              }
            ]
          }
        ]
      },
      "l3v6": {
        "rule": [
          {
            "rule-name": "rule2",
            "ordinal": [
              {
                "ordinal-value": 2,
                "match": {
                  "direction": "ingress",
                  "destination-ipv6-local": "true"
                },
                "action": {
                  "drop": "true"
                }
              }
            ]
          }
        ]
      }
    }
  }
```

**Configuring ACL Actions**

**Syntax:**

**set forwarding-options acl [l3v4 | l3v6] rule** <rulename> **ordinal** <ordinal_value> **action** <attribute> <value>

| Attribute | Description |
|---|---|
| `drop true` | If the ACL rule specifies `drop true`, the system will discard any packets that match that rule. Use the `delete` form of the command for the system to ignore the rule. |
| `permit true` | If the ACL rule specifies `permit true`, the system forwards traffic matching that rule. Use the `delete` form of the command for the system to ignore the rule. |
| `action statistics true` | Configure action, enable statistics. Use the `delete` form of the command to disable the configuration.<br><br>A limited number of counter resources are available in a common pool for user-defined ACLs, protocol ACLs, L3, and L2X logical interfaces. |
| `forward-class <class>` | Specifies forward class value (class-0 to class-7, class-all)<br><br>The user-defined action `forward-class` is supported only on the UfiSpace S9510-28DC and Edgecore CSR440 platforms. |
| `mirror <mirror>` | Specifies ACL action mirror name.<br><br>Currently, ACLs with mirror actions are not supported. |
| `capture true` | You can enable the capture action when using the RBFS built-in capture feature with an ACL to more granularly specify the traffic to be captured. To disable, use the `delete` form of the command.<br><br>NOTE: For more information and an example, refer to the *RBFS NOC Troubleshooting Guide*. |
| `policer-name <policer-name>` | Specifies policer profile name. |
| `redirect-to-cpu true` | Configure action, redirect packets to CPU. To disable, use the `delete` form of the command. |
| `action traffic-sample true` | Allows you to sample traffic that matches specific ACL rules and mirror it to the CPU for analysis. |
| `trap default` | Allows the system to trap matching packets to the control plane and process them. |

API To access the RESTCONF API that corresponds to this CLI, click here.

Example 1: ACL Rule with Drop Action

```
{
```

```
            "rule-name": "rtb_firewall_two",
            "ordinal": [
              {
                "ordinal-value": 1000,
                "match": {
                  "direction": "ingress",
                  "source-ipv4-prefix": "198.51.100.50/24",
                  "source-l4-port": 8080
                },
                "action": {
                  "drop": "true"
                }
              }
            ]
          }
```

Example 2: ACL-based Traffic Sampling

```
set forwarding-options acl l3v4 rule sample-icmp ordinal 100 action traffic-sample
true
```

```
{
  "forwarding-options": {
    "acl": {
      "l3v4": {
        "rule": [
          {
            "name": "sample-icmp",
            "ordinal": 100,
            "action": {
              "traffic-sample": true
            }
          }
        ]
      }
    }
  }
}
```

After the ACL is configured with the traffic-sample action, you can use the `capture interface <intf_name>` CLI to monitor traffic that matches the specified ACL.

#### Configuring Prefix Lists

#### Configuring IPv4/IPv6 Prefix List for ACL and Multifield Classifier

**Syntax:**

**set forwarding-options prefix-list** <prefix-list-name> <attribute> <value>

| Attribute | Description |
|---|---|
| `<prefix-list-name>` | Name of the prefix list, which will be later used to attach with ACL configuration. |
| `ipv4-prefix <ipv4_prefix>` | Specifies the IPv4 prefix address. |
| `ipv6-prefix <ipv6_prefix>` | Specifies the IPv6 prefix address. |

API To access the RESTCONF API that corresponds to this CLI, click here.

**Example 1: Prefix List Configuration**

```
supervisor@rtbrick>LEAF01: op> show config forwarding-options prefix-list
{
    "rtbrick-config:prefix-list": [
      {
        "prefix-list-name": "ipv4-list",
        "ipv4-prefix": [
          {
            "ipv4-prefix": "198.51.100.50/24"
          },
          {
            "ipv4-prefix": "198.51.101.60/24"
          },
          {
            "ipv4-prefix": "198.51.102.70/24"
          }
        ]
      }
    ]
  }
```

**Example 2: Using Prefix-list in Multifield-Classifier**

```
supervisor@rtbrick>LEAF01: op> show config forwarding-options prefix-list pta-iptv-
multicast
{
  "rtbrick-config:prefix-list": [
    {
      "prefix-list-name": "ipv4-list",
      "ipv4-prefix": [
          {
            "ipv4-prefix": "198.51.100.50/24"
          },
          {
            "ipv4-prefix": "198.51.101.60/24"
```

```
          },
          {
            "ipv4-prefix": "198.51.102.70/24"
          }
      ]
    }
  ]
}
```

**Example 3: Viewing Multifield-Classifier Details**

```
supervisor@rtbrick>LEAF01: op> show config forwarding-options class-of-service
multifield-classifier acl l3v4 rule pta-triple-play-8queues ordinal 6000
{
  "rtbrick-config:ordinal": [
    {
      "ordinal-value": 6000,
      "match": {
        "destination-ipv4-prefix-list": "ipv4-list"
      },
      "action": {
        "forward-class": "class-1",
        "remark-codepoint": 248
      }
    }
  ]
}
```

### 6.3.3. ACL Operational Commands

**ACL Show Commands**

**Syntax:**

**show acl** <option>

| Option | Description |
|---|---|
| `-` | Without any option, this command displays brief information about the access control list (ACL). |
| `all` | Displays all ACL information including the internal ACLs. |
| `all count` | Displays the count of all ACL rules. |
| `all detail` | Displays detailed information for all ACLs. |
| `all ordinal` | Displays detailed ordinal information for all ACLs. In addition, you can display ACL information based on `rule` and `type` for for the specified ordinal. |

| Option | Description |
|---|---|
| all rule <acl-name> | Displays detailed information for the specified ACL rule name. In addition, you can display ACL information based on ordinal and type for the specified rule. |
| count | Displays the count of ACL rules. |
| detail | Displays detailed information about the access-control list (ACL). |
| type <acl_type> | Displays detailed information for the specified ACL type. |
| rule <acl-rule-name> | Displays detailed information for the specified ACL rule name. |
| ordinal <ordinal> | Displays detailed information for the specified Ordinal. |
| type <acl_type> rule <rule_name> | Displays detailed information for the specified ACL Type and Rule name. |
| type <acl_type> ordinal <ordinal> | Displays detailed information for the specified ACL Type and Ordinal. |
| rule <rule_name> type <acl_type> | Displays detailed information for the specified ACL Type and Rule name. |
| rule <rule_name> ordinal <ordinal> | Displays detailed information for the specified Rule name and Ordinal. |
| ordinal <ordinal> type <acl_type> | Displays detailed information for the specified ACL type and ordinal. |
| ordinal <ordinal> rule <rule_name> | Displays detailed information for the specified rule name and ordinal. |
| type <acl_type> rule <rule_name> ordinal <ordinal> | Displays detailed information for the specified Ordinal, ACL type and rule name. |
| type <acl_type> ordinal <ordinal> rule <rule_name> | Displays detailed information for the specified Ordinal, ACL type and rule name. |
| rule <rule_name> ordinal <ordinal> type <acl_type> | Displays detailed information for the specified Ordinal, ACL type and rule name. |
| rule <rule_name> type <acl_type> ordinal <ordinal> | Displays detailed information for the specified Ordinal, ACL type and rule name. |
| ordinal <ordinal> rule <rule_name> type <acl_type> | Displays detailed information for the specified Ordinal, ACL type and rule name. |
| ordinal <ordinal> type <acl_type> rule <rule_name> | Displays detailed information for the specified Ordinal, ACL type and rule name. |

Example 1: Show information about ACLs

```
supervisor@rtbrick: op> show acl | head -n 10
ACL                                      Ordinal  Type                Attach Point     Priority   Policer Name
_DEFAULT_MTU_POLICER_TRAP_ACL                  -  l2                  recycle-0/0/211         1   _DEFAULT_POLICER_20_MB
recycle-0/0/215_bypass_acl                     -  l2                  recycle-0/0/215         2   -
lldp.ifp-0/1/12.trap.rule.0                    -  l2                  ifp-0/1/12             50
_DEFAULT_POLICER_LLDP_GLOBAL_10MB
```

```
lldp.ifp-0/1/12.trap.rule.1                  -  l2                ifp-0/1/12            50
_DEFAULT_POLICER_LLDP_GLOBAL_10MB
lldp.ifp-0/1/12.trap.rule.2                  -  l2                ifp-0/1/12            50
_DEFAULT_POLICER_LLDP_GLOBAL_10MB
lldp.ifp-0/1/32.trap.rule.0                  -  l2                ifp-0/1/32            50
_DEFAULT_POLICER_LLDP_GLOBAL_10MB
lldp.ifp-0/1/32.trap.rule.1                  -  l2                ifp-0/1/32            50
_DEFAULT_POLICER_LLDP_GLOBAL_10MB
lldp.ifp-0/1/32.trap.rule.2                  -  l2                ifp-0/1/32            50
_DEFAULT_POLICER_LLDP_GLOBAL_10MB
lldp.ifp-0/1/24.trap.rule.0                  -  l2                ifp-0/1/24            50
_DEFAULT_POLICER_LLDP_GLOBAL_10MB
```

Example 2: Show detailed information about ACLs

```
supervisor@rtbrick: op> show acl detail | head -n 10
Rule: _DEFAULT_MTU_POLICER_TRAP_ACL
  ACL type: l2
  Ordinal: -             Priority: 1
    Match:
      Attachment point: recycle-0/0/211
      Direction: ingress
      Physical Interface: recycle-0/0/211
    Action:
      Redirect to CPU: True
      Policer profile name: _DEFAULT_POLICER_20_MB
```

Example 3: Show detailed information for a specified ACL Rule

```
supervisor@rtbrick: op> show acl rule DROP_VIDEO_PKTS
Rule: DROP_VIDEO_PKTS
  ACL type: ext_l3v4
  Ordinal: 1             Priority: 500
    Match:
      Direction: external
      Destination IPv4 prefix: 192.0.2.54/32
    Action:
      Stats enabled: True
    Result:
      Trap ID: user-defined
      ACL Handle: 4
      External:
        Value: 2763663
          Is Trap ID: Yes
          Trap ID: 199
          Trap Strength: 5
          Is QoS: Yes
          Class: 0
          Policer ID: 21
```

Example 4: Show detailed information for a specified ACL Type and Rule name

```
supervisor@rtbrick: op> show acl all
ACL                                     Ordinal  Type                Attach Point     Priority   Policer Name
Acl Handle
radius-srv1-v4-coa-trap                       -  ext_l3v4            -                     50
_DEFAULT_POLICER_RADIUS_GLOBAL_...          5
radius-srv1-v4-acct-trap                      -  ext_l3v4            -                     50
_DEFAULT_POLICER_RADIUS_GLOBAL_...          4
mpls.oam.destination.ipv4.trap.rule           -  ext_l3v4            -                     50
_DEFAULT_POLICER_OAM_GLOBAL_20MB            1
radius-srv1-v4-auth-trap                      -  ext_l3v4            -                     50
_DEFAULT_POLICER_RADIUS_GLOBAL_...          3
mpls.oam.source.ipv4.trap.rule                -  ext_l3v4            -                     50
_DEFAULT_POLICER_OAM_GLOBAL_20MB            2
_DEFAULT_MTU_POLICER_TRAP_ACL                 -  l2                  recycle-0/0/211        1   _DEFAULT_POLICER_20_MB
26
l2x_xconnect_mpls_port_wide_lif_data_acl     10  l2                  -                      2   -
13
arp_acl_sc_double_tag                         -  l2                  -                      4
_DEFAULT_POLICER_ARP_GLOBAL_10MB           24
arp_acl_s_single_tag                          -  l2                  -                      4
_DEFAULT_POLICER_ARP_GLOBAL_10MB           25
arp_acl_single_tag                            -  l2                  -                      4
_DEFAULT_POLICER_ARP_GLOBAL_10MB           22
arp_acl_double_tag                            -  l2                  -                      4
_DEFAULT_POLICER_ARP_GLOBAL_10MB           23
arp_acl_untag                                 -  l2                  -                      4
_DEFAULT_POLICER_ARP_GLOBAL_10MB           21
lldp.ifp-0/1/31.trap.rule                     0  l2                  ifp-0/1/31            50

<...>
```

Example 5: Show the count of all ACL rules

```
supervisor@rtbrick: op> show acl all count
ACL count
  ext_l3v4                  : 5
  l2                        : 55
  PPPOE                     : 8
  service_rule              : 4
  ext_l3v6                  : 7
  l2tp_cross_connect        : 1
  l2tp_options              : 1
  Total-count               : 81
```

### ACL Statistics Commands

ACL statistics are currently not supported for PIM, IGMP, and L2TP protocol traffic.

**Syntax:**

**show acl** <option> **statistics**

| Option | Description |
|---|---|
| statistics | Displays ACL statistics information |
| <acl-name> statistics | Displays ACL statistics information for the specified ACL. |
| type <acl_type> statistics | Displays ACL statistics information for the specified ACL type. |
| rule <rule_name> statistics | Displays ACL statistics information for the specified rule name. |
| ordinal <ordinal> statistics | Displays ACL statistics information for the specified ordinal. |
| type <acl_type> rule <rule_name> statistics | Displays ACL statistics information for the specified ACL type and rule name. |
| type <acl_type> ordinal <ordinal> statistics | Displays ACL statistics information for the specified ACL type and ordinal. |
| rule <rule_name> type <acl_type> statistics | Displays ACL statistics information for the specified ACL type and rule name. |
| rule <rule_name> ordinal <ordinal> statistics | Displays ACL statistics information for the specified rule name and ordinal. |
| ordinal <ordinal> type <acl_type> statistics | Displays ACL statistics information for the specified ACL type and ordinal. |
| ordinal <ordinal> rule <rule_name> statistics | Displays ACL statistics information for the specified rule name and ordinal. |
| <acl_type> rule <rule_name> ordinal <ordinal> statistics | Displays ACL statistics information for the specified ordinal, ACL type and rule name. |
| type <acl_type> ordinal <ordinal> rule <rule_name> statistics | Displays ACL statistics information for the specified ordinal, ACL type and rule name. |
| rule <rule_name> ordinal <ordinal> type <acl_type> statistics | Displays ACL statistics information for the specified ordinal, ACL type and rule name. |
| rule <rule_name> type <acl_type> ordinal <ordinal> statistics | Displays ACL statistics information for the specified ordinal, ACL type and rule name. |
| ordinal <ordinal> rule <rule_name> type <acl_type> statistics | Displays ACL statistics information for the specified ordinal, ACL type and rule name. |
| ordinal <ordinal> type <acl_type> rule <rule_name> statistics | Displays ACL statistics information for the specified ordinal, ACL type and rule name. |

Example 1: Display ACL statistics information

```
supervisor@rtbrick: op> show acl statistics |head -n 10
ACL                                      Ordinal    Type              Units      Total       Accepted    Dropped
lldp.ifp-0/1/12.trap.rule.0              -          l2                Packets    59          59          0
                                                                      Bytes      7316        7316        0
lldp.ifp-0/1/12.trap.rule.1              -          l2                Packets    59          59          0
                                                                      Bytes      7316        7316        0
lldp.ifp-0/1/12.trap.rule.2              -          l2                Packets    59          59          0
```

```
                                                                Bytes      7316        7316        0
lldp.ifp-0/1/32.trap.rule.0              -         l2           Packets    59          59          0
                                                                Bytes      7316        7316        0
lldp.ifp-0/1/32.trap.rule.1              -         l2           Packets    59          59          0
```

Example 2: Display ACL statistics information for the specified ACL

```
supervisor@rtbrick: op> show acl rule DROP_VIDEO_PKTS statistics
ACL                                      Ordinal    Type         Units      Total       Accepted    Dropped
DROP_VIDEO_PKTS                          1          ext_l3v4     Packets    100         0           100
                                                                 Bytes      9200        0           9200
```

### Show Trap Commands

A trap refers to a mechanism where specific packets or traffic conditions are intercepted (or "trapped") by the system—typically from the data plane to the control plane—for further processing. It provides visibility into how traffic is handled by the system. It includes both forwarded and dropped packet statistics, helping operators analyze traffic behavior to identify anomalies.

**Syntax:**

**show trap** <option>

| Option | Description |
|---|---|
| `trap_name` | Specifies the name of the trap. Displays detailed information about a specific trap, including its configuration, state, and associated parameters. |
| `statistics` | Displays statistical information for all traps, such as the number of traps generated, sent, or failed. |

Example for trap statistics of RPF.

```
supervisor@rtbrick.net: cfg> show trap statistics rpf
Trap Statistics Type: rpf
    Statistics ID: 1
    Counters:
        Forward-Packets: 0
        Forward-Bytes: 0
        Drop-Packets: 1550
        Drop-Bytes: 2194800
```

Example for trap statistics of NAT.

```
supervisor@rtbrick.net: cfg> show trap statistics nat
Trap Statistics Type: nat
```

```
    Statistics ID: 524
    Counters:
        Forward-Packets: 20943738
        Forward-Bytes: 29405223462
        Drop-Packets: 38998908
        Drop-Bytes: 54754548576
```

### ACL Clear Commands

Clear commands allow resetting operational states.

#### Clear ACL Statistics

**Syntax:**

**clear acl** <options>

| Option | Description |
|---|---|
| `statistics` | Clears all the ACL statistics. |
| `type <acl_type> statistics` | Clears all ACL statistics for the specified ACL type |
| `rule <rule_name> statistics` | Clears all ACL statistics for the specified rule |
| `ordinal <ordinal> statistics` | Clears all ACL statistics for the specified ordinal |
| `type <acl_type> rule <rule_name> statistics` | Clears all ACL statistics for the specified ACL type and rule |
| `type <acl_type> ordinal <ordinal> statistics` | Clears all ACL statistics for the specified ACL type and ordinal |
| `rule <rule_name> type <acl_type> statistics` | Clears all ACL statistics for the specified ACL type and rule |
| `rule <rule_name> ordinal <ordinal> statistics` | Clears all ACL statistics for the specified rule and ordinal |
| `ordinal <ordinal> type <acl_type> statistics` | Clears all ACL statistics for the specified ACL Type and Ordinal |
| `ordinal <ordinal> rule <rule_name> statistics` | Clears all ACL statistics for the specified Rule name and Ordinal |
| `type <acl_type> rule <rule_name> ordinal <ordinal> statistics` | Clears all ACL statistics for the specified Ordinal, ACL Type and Rule name |
| `type <acl_type> ordinal <ordinal> rule <rule_name> statistics` | Clears all ACL statistics for the specified Ordinal, ACL Type and Rule name |
| `rule <rule_name> ordinal <ordinal> type <acl_type> statistics` | Clears all ACL statistics for the specified Ordinal, ACL Type and Rule name |
| `rule <rule_name> type <acl_type> ordinal <ordinal> statistics` | Clears all ACL statistics for the specified Ordinal, ACL Type and Rule name |

| Option | Description |
| --- | --- |
| `ordinal <ordinal> rule <rule_name> type <acl_type> statistics` | Clears all ACL statistics for the specified Ordinal, ACL Type and Rule name |
| `ordinal <ordinal> type <acl_type> rule <rule_name> statistics` | Clears all ACL statistics for the specified Ordinal, ACL Type and Rule name |

Example: Clearing ACL Statistics of a specified ACL Rule

```
supervisor@rtbrick>LEAF01: op> clear acl rule lldp.ifp-0/0/44.trap.rule statistics
Success : command success
supervisor@rtbrick>LEAF01: op>
```

#### Clear Trap Statistics

This command clears the accumulated statistics for all traps.

**Syntax:**

```
clear trap statistics
```

Example:

```
clear trap statistics rpf
```

```
supervisor@rtbrick.net: cfg> clear trap statistics rpf
Success : command success
```

# 6.4. Port Mirroring

## 6.4.1. Port Mirroring Overview

Port Mirroring is a method of monitoring network traffic. When you enable port mirroring, the switch sends a copy of all network packets seen on one port to another port, where the packet can be analyzed.

#### Inbound Mirroring

Inbound mirroring is defined per In-Port, or per In-Port x VLAN. Configurations for six distinct VLAN tags, for any other VLAN tag, and for packets without VLAN tags are supported. The ingress mirroring can be sampled by specifying a probability that a matching packet will be mirrored.

#### Outbound Mirroring

Outbound mirroring is defined per Out-Port, or per Out-Port x VLAN tag. Configurations for seven distinct VLAN tags are supported.

> Only configure one, either the IFL or the IFP, as the mirror source interface. Do not configure both a source IFL and a source IFP for the same mirror as it causes hardware failure.

#### Guidelines and Limitations

- Up to 15 mirror profiles can be configured.
- The same mirror resources are used for Lawful Interception (LI) and Port Mirroring.
- You can configure a CPU port as destination physical interface port; but if heavy traffic is mirrored, it may impact system performance.
- If physical interface/logical interface goes down, mirror configuration will not be deleted automatically. You need to delete the mirror configuration explicitly.
- Before creating logical interface mirroring, the source logical interface should exist.
- The logical interface should not be deleted during mirroring.
- If you want to mirror traffic to CPU, enable the control plane security features. For more refer, see the *Control Plane Security Guide*.
- Since this is a debugging tool, the save and reload functionality is not supported.

> Enabling port mirroring impacts CPU performance. This performance degradation occurs as soon as the mirror is configured, depending on active traffic volume.

#### Supported Platforms

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

### 6.4.2. Port Mirroring Configuration

#### Configuration Hierarchy

The diagram illustrates the port mirroring configuration hierarchy.

*Figure 117. Port Mirroring Configuration Hierarchy*

### Configuration Syntax and Commands

The following sections describe the port mirroring configuration syntax and commands.

**Syntax:**

**set forwarding-options mirror** <name> [**source** | **destination**] <attribute> <value>

| Attribute | Value |
|---|---|
| `<name>` | Name for mirror configuration |
| `acl true` | Configure source as ACL.<br>Use `delete` form of the command to disable the configuration. |
| `direction [egress \| ingress]` | Configure traffic direction ingress/egress. |
| `interface <interface>` | Specifies the physical interface name. |
| `logical-interface <logical-interface>` | Configure source logical interface name. |

Example 1: Mirroring one physical interface traffic to another physical interface

```
{
    "rtbrick-config:mirror": [
      {
        "name": "MIRROR1",
        "destination": {
          "interface": "ifp-0/0/4"
        },
        "source": {
          "direction": "ingress",
          "interface": "ifp-0/0/2"
        }
      }
    ]
  }
```

Example 2: Mirroring Traffic to CPU

```
{
    "rtbrick-config:mirror": [
      {
        "name": "mirror1",
        "destination": {
          "interface": "cpu-0/0/200"
        },
```

```
        "source": {
          "direction": "ingress",
          "interface": "ifp-0/0/52"
        }
      }
    ]
  }
```

### 6.4.3. Port Mirroring Operational Commands

#### Capturing Mirror Traffic

After you configure mirroring to CPU by using the commands above, you can use the **capture** command to capture the mirror traffic.

**Syntax:**

**capture mirror file** <file_name> [**start** | **stop**]

| Attribute | Description |
|---|---|
| `<file_name>` | Name of the file where mirror traffic is captured |

Example: Starting and stopping mirror traffic to a file

```
capture mirror file test.pcap start
capture mirror file test.pcap stop
```

#### Show Commands

##### show capture sessions

**Syntax:**

**show capture sessions**

| Option | Description |
|---|---|
| `-` | Without any option, the commands displays the FIB packet Capture sessions. |

Example: Summary of FIB packet capture sessions

```
supervisor@rtbrick: op> show capture sessions
Interface                      Direction   File                            Context
```

```
ifp-0/1/23                    BOTH        file1.pcap                    -
```

# 6.5. LAG

## 6.5.1. LAG Overview

A link aggregation group (LAG) combines multiple physical links into a single logical interface which is referred to as a bundle interface. These physical links are connected between two devices. The device uses LACP protocol to bundle the member links and create high speed connections. Although a bundle can be created based on static configuration, bandwidth can be increased by adding member links to the bundle. This also allows load sharing among the physical links. Thus, a group of ports combined together is called a link aggregation group, or LAG.

The LAG interface combines the bandwidth of the individual member links. The properties like speed and bandwidth of the individual member links should be the same to make it part of that LAG. The traffic which is directed towards the LAG interface is sent on the individual member links. This traffic is not pinned to a specific member link but rather determined by a specific flow. This hash could be calculated based on various fields in the packet.

#### LAG Interface Modes

The LAG interface could be formed statically or dynamically. LACP protocol helps to bring up the interface dynamically. The two modes of LAG interface are:

1. **Static LAG**: In this mode, the member links do not initiate or process any of the LACP packets received. The device brings up the LAG interface without LACP negotiation.
2. **Dynamic LAG**: In this mode, the member links process the LACP packets received. Under this mode, there two sub modes:
   a. **active**: LACP packets are generated on each of the member links on the transmit side.
   b. **passive**: LACP packets are generated on the member link in response to the LACP packet received. That means, at least one side of the LAG should be configured as `active` to bring up the LAG interface.

#### Layer2 and Layer 3 Interfaces

LAG interfaces can be used as layer 2 and layer 3 interfaces. A regular layer 2 or layer 3 interface can be created on top of the single LAG interface. These interfaces can be divided based on 802.1q VLAN ID's. Multiple layer 3 interfaces can be created and each of them can be associated with different instances.

#### LACP (Link Aggregation Control Protocol)

LACP is part of an IEEE specification (802.3ad) that allows several physical ports to be grouped to form a single logical interface. LACP allows a switch to negotiate an LAG by sending LACP packets on its member links. It negotiates the various configuration parameters to bring up the individual member links.

### Interface Holddown

The Interface Holddown feature remains supported when interfaces are grouped in a LAG. For more information about the Interface Holddown feature, see Interface Holddown.

### Configurable State for LAG Member Interfaces

RBFS provides the `member-follow-lag-admin-state` command option for link aggregation interface configuration to control the state of member interfaces. By default, member interfaces go to the 'admin-down' state when the LAG interface is disabled. With this option enabled, member interfaces can remain operationally independent and are not forced to admin-down when the LAG interface is disabled.

### Guidelines and Limitations

- You cannot configure logical interfaces on a LAG member ports.
- You cannot configure L2X on a LAG member port.

### LAG Hashing Algorithm

The LAG hashing algorithm distributes traffic across multiple paths or links for load balancing. It achieves this by using various packet header fields (like IP addresses, port numbers, or MAC addresses) to generate a hash value, ensuring probability and an equitable distribution of network traffic.

The following table lists the packet header fields that are used by the LAG hashing algorithm to generate a hash value, which in turn determines how traffic is distributed. A tick (✔) indicates that the field is used by the hashing algorithm. A cross mark (✘) indicates that the field is not used by the hashing algorithm.

#### IPv4 UDP/TCP

The following table lists the header fields used in the LAG hashing algorithm for different platforms when handling IPv4 UDP/TCP traffic.

| Header Field | UfiSpace S9600-102XC | UfiSpace S9600-72XC | UfiSpace S9600-32X | UfiSpace S9510-28DC | UfiSpace S9500-22XST | Edgecore AGR420 | Edgecore AGR400 | Edgecore CSR440 |
|---|---|---|---|---|---|---|---|---|
| Source IP Address (SIP) | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Destination IP Address (DIP) | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Protocol | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Source Port | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Destination Port | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |

#### IPv4

The following table lists the header fields used in the LAG hashing algorithm for different platforms when handling IPv4 traffic.

| Header Field | UfiSpace S9600-102XC | UfiSpace S9600-72XC | UfiSpace S9600-32X | UfiSpace S9510-28DC | UfiSpace S9500-22XST | Edgecore AGR420 | Edgecore AGR400 | Edgecore CSR440 |
|---|---|---|---|---|---|---|---|---|
| Source IP Address (SIP) | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Destination IP Address (DIP) | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Protocol | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |

#### IPv6 UDP/TCP

The following table lists the header fields used in the LAG hashing algorithm for different platforms when handling IPv6 UDP/TCP traffic.

| Header Field | UfiSpace S9600-102XC | UfiSpace S9600-72XC | UfiSpace S9600-32X | UfiSpace S9510-28DC | UfiSpace S9500-22XST | Edgecore AGR420 | Edgecore AGR400 | Edgecore CSR440 |
|---|---|---|---|---|---|---|---|---|
| Source IP Address (SIP) | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Destination IP Address (DIP) | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Next Header | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Source Port | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Destination Port | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Flow Label (FL) | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |

#### IPv6

The following table lists the header fields used in the LAG hashing algorithm for different platforms when handling IPv6 traffic.

| Header Field | UfiSpace S9600-102XC | UfiSpace S9600-72XC | UfiSpace S9600-32X | UfiSpace S9510-28DC | UfiSpace S9500-22XST | Edgecore AGR420 | Edgecore AGR400 | Edgecore CSR440 |
|---|---|---|---|---|---|---|---|---|
| Source IP Address (SIP) | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Destination IP Address (DIP) | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Next Header | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |

| Header Field | UfiSpace S9600-102XC | UfiSpace S9600-72XC | UfiSpace S9600-32X | UfiSpace S9510-28DC | UfiSpace S9500-22XST | Edgecore AGR420 | Edgecore AGR400 | Edgecore CSR440 |
|---|---|---|---|---|---|---|---|---|
| Flow Label (FL) | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |

#### Ethernet (L2 Forwarding)

The following table lists the header fields used in the LAG hashing algorithm for different platforms when handling Ethernet (L2 Forwarding) traffic.

| Header Field | UfiSpace S9600-102XC | UfiSpace S9600-72XC | UfiSpace S9600-32X | UfiSpace S9510-28DC | UfiSpace S9500-22XST | Edgecore AGR420 | Edgecore AGR400 | Edgecore CSR440 |
|---|---|---|---|---|---|---|---|---|
| Source MAC | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Destination MAC | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| EtherType | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| VLAN | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |

#### MPLS

The following table lists the header fields used in the LAG hashing algorithm for different platforms when handling MPLS traffic.

| Header Field | UfiSpace S9600-102XC | UfiSpace S9600-72XC | UfiSpace S9600-32X | UfiSpace S9510-28DC | UfiSpace S9500-22XST | Edgecore AGR420 | Edgecore AGR400 | Edgecore CSR440 |
|---|---|---|---|---|---|---|---|---|
| All MPLS Labels | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |

#### Field Definitions

**SIP/DIP (Source/Destination IP Address)**

Identifies the source/destination of the packet, that is, IPv4/IPv6

**Next Header/Protocol**

Identifies the type of payload, such as TCP, UDP etc.

**L4 Ports (Source/Destination Port)**

Used in TCP/UDP headers to identify application endpoints

**FL (Flow Label)**

Used for IPv6 to improve load balancing

**MAC Addresses (Source/Destination MAC)**

Used in Ethernet frames for identifying source and destination devices

**EtherType**

Indicates the protocol encapsulated in the Ethernet frame

**VLAN**

Identifies the VLAN to which the frame belongs

**MPLS Labels**

Used in MPLS networks to identify the destination / Forward Equivalence Class for packet forwarding

#### Supported Platforms

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

#### Supported Number of LAG Interfaces on Platforms

For information on the maximum number of LAG interfaces and members per LAG supported on each hardware platform, refer to the Supported Platforms section of the *Platform Guide*.

### 6.5.2. LAG Configuration

#### Creating LAG Interfaces

**Syntax**:

**set link-aggregation interface** <name> <attribute> <value>

| Attribute | Description |
|---|---|
| `<name>` | Specifies the name of the LAG interface. The supported LAG interface names: 'lag-1' to 'lag-99'. |
| `<description>` | Link aggregation interface description |
| `mode <mode>` | Specifies the LAG mode. The default mode is LACP. The possible modes are:<br>• lacp - In this mode, the member links processes LACP packets received. When you create a LAG interface in LACP mode, the LACP PDUs are sent and received through member interfaces.<br>• static: In this mode, the member links do not initiate or process any of the LACP packets received. |
| `<minimum-link-count>` | Specify the minimum number of active member links required for the link aggregation interface. |
| `<member-interface>` | Specify name of the member interface. |

| Attribute | Description |
| --- | --- |
| `redundancy-session-id` | Specify the value for the redundancy group session ID. Range from 1 to 65535 is allowed. |
| `system-id` | Specify the MAC address (as system ID) of the device for the link-aggregation interface. |

`redundancy-session-id`, and `system-id` attributes can only be used when you deploy RBFS in redundancy mode. For information about LAG configuration when deploying RBFS in redundancy mode, see RBFS Redundancy Solution Guide.

**Example: LAG Interfaces Configuration**

```
supervisor@rtbrick: cfg> show config link-aggregation
{
    "rtbrick-config:link-aggregation": {
      "interface": [
        {
          "interface-name": "lag-3",
          "mode": "lacp",
          "minimum-link-count": 2,
          "member-interface": [
            {
              "member-interface-name": "ifp-0/0/1",
              "lacp-mode": "active"
            },
            {
              "member-interface-name": "ifp-0/0/5",
              "lacp-mode": "active"
            }
            {
              "member-interface-name": "ifp-0/0/5",
              "lacp-mode": "passive"
            }
          ]
        }
      ]
    }
  }
```

To access the RESTCONF API that corresponds to this CLI, click here.

**Configuring LAG Member Interfaces**

You can add member ports to the LAG interface. The command below allows you to bundle multiple physical interfaces with similar properties like speed, MTU, MRU.

**Syntax**:

**set link-aggregation interface** <name> **member-interface** <name> <attribute> <value>

| Attribute | Description |
|---|---|
| `lacp-mode <mode>` | Specifies the LACP mode. The default lacp-mode is `active`.<br>**active**: LACP packets are generated on each of the member links on the trad, the receive side.<br>**passive**: LACP packets are generated on the member link in response to the LACP packet received at one side of the LAG should be configured as `active` to bring the LAG interface. |
| `lacp-timeout <timeout-value>` | Specifies the timeout for the LACP session. A long timeout is 90 seconds, while a short is 3 seconds (default is short). Setting the timeout value will instruct the partner at which interval it should send the updates (30 seconds for long timeout, 1 second for short timeout). |
| `member-follow-lag-admin-state false` | Enables independent operational control for individual LAG member ports.<br><br>NOTE: Having mismatching timeouts will not break the operation, even though it is not desirable design-wise. This is because in LACP both Actor and Partner negotiate the transmission rate, so the transmitter sends at the receiver's expected interval. |

#### Configuring State for LAG Member Interfaces

By default, member interfaces inherit the administrative state of the LAG. Setting the option to `false` allows member interfaces to maintain their own administrative state independently of the LAG. To restore the default behavior, use the `delete` form of the command.

**Syntax**:

**set link-aggregation interface** <interface-name> **member-follow-lag-admin-state false**

The following table provides command options and descriptions.

| Attribute | Description |
|---|---|
| `<interface-name>` | Name of the LAG interface. |
| `member-follow-lag-admin-state false` | Enables independent operational control for individual LAG member ports. |

This configuration creates LAG interface 'lag-1' and enables the mode LACP mode for dynamic link aggregation. The `member-follow-lag-admin-state` option is set to false, which means member interfaces do not automatically follow the administrative state of the LAG. The LAG includes a single member interface 'ifp-0/0/16'.

```
set link-aggregation interface lag-1
set link-aggregation interface lag-1 mode lacp
set link-aggregation interface lag-1 minimum-link-count 1
set link-aggregation interface lag-1 member-follow-lag-admin-state false
set link-aggregation interface lag-1 member-interface ifp-0/0/16
```

```
supervisor@rtbrick.net: cfg> show conf link-aggregation
{
  "rtbrick-config:link-aggregation": {
    "interface": [
      {
        "interface-name": "lag-1",
        "mode": "lacp",
        "minimum-link-count": 1,
        "member-follow-lag-admin-state": "false",
        "member-interface": [
          {
            "member-interface-name": "ifp-0/0/16"
          }
        ]
      }
    ]
  }
}
```

**Configuring QoS on LAG Interface**

RBFS supports QoS at physical interface level for LAG. Users can apply QoS profile at physical interface level through which one common QoS classification can be applied for all traffic on that port, irrespective of the destination logical interface.

The following features are supported:

- Classification (IEEE-802.1)
- Remarking (IEEE-802.1)
- Ingress Policing
- Egress Policing

For information about configuring the above features, refer the *HQoS Configuration Guide*.

- You cannot apply QoS class of service on LAG logical interface
- Currently, queuing and scheduling are not supported

**Syntax**:

**set interface** <physical interface> **class-of-service** <class-of-service>

| Attribute | Description |
|---|---|
| `<interface>` | Name of the interface |
| `<class-of-service>` | Specifies the class of service |

**Example**:

```
supervisor@rtbrick: cfg> set interface lag-11 class-of-service Retail_profile
supervisor@rtbrick: cfg> commit
supervisor@rtbrick: cfg> show config int lag-11
{
    "rtbrick-config:interface": [
      {
        "name": "lag-11",
        "class-of-service": "Retail_profile"
      }
    ]
  }
```

**Configuring L2X on LAG Interface**

All forms of L2X that are supported on the regular physical interfaces are supported on LAG. The incoming packet is be matched to a specific L2X profile based on the Cross Connect configuration on the specified LAG interface.

The following match conditions are supported on the LAG interface:

- Incoming LAG interface without any VLAN
- Incoming LAG interface with a single VLAN
- Incoming LAG interface with inner and Outer VLAN
- Incoming LAG interface with any single VLAN
- Incoming LAG interface with inner VLAN and any outer VLAN

For information about configuring L2X, see the L2X Configuration Guide.

The following table provides the L2X match action attributes which are supported on LAG interface.

| Attribute | Description |
|---|---|
| `nexthop6 <nexthop>` | Next-Hop address |
| `match-type <match-type>` | Match types with which traffic can be matched. |

| Attribute | Description |
|---|---|
| service-label <service_label> | Service label value. NOTE: Supported MPLS label values are 0 - 1048575. The reserved MPLS label range is 0 - 15. In RBFS, BGP uses the label range 20000 - 100000. It is recommended to assign label values outside of these reserved ranges to avoid conflicts. |
| ingress-vlan-operation <ingress-vlan-action> | VLAN operation on ingress side outer VLAN |
| ingress-outer-vlan <vlan-id> | Outer VLAN at ingress side |
| outgoing_ifp | Outgoing interface |
| vlan_operation | VLAN operation |
| outgoing_outer_vlan1 | Outgoing outer VLAN |

### Configuring Holddown on LAG Interface

By default, the Interface Hold-down feature is enabled. You can disable the feature if you want. It is allowed to configure timer value for options such as 'holddown down delay' and 'holddown up delay'.

**Syntax:**

**set interface** <interface-name> <attribute> <value>

| Attribute | Description |
|---|---|
| <interface-name> | Name of the lag. Example: lag-1. |
| holddown <disable\|enable> | Enable or disable the holddown timer feature. By default, the feature is enabled. |
| holddown-down-delay | Configure the down delay time in milliseconds (ms). The allowed values include 0, 100, 200, 500, 1000, 2000, 5000, and 10000. It overrides the default delay of '0' milliseconds. |
| holddown-up-delay | Configure the up delay time milliseconds (ms). The allowed values include 0, 100, 200, 500, 1000, 2000, 5000, and 10000. |

Example Commands:

```
set interface lag-1
set interface lag-1 holddown-down-delay 1000
set interface lag-1 holddown-up-delay 5000
```

Example:

```
supervisor@rtbrick.net: cfg> show config  interface lag-1
{
  "rtbrick-config:interface": [
    {
      "name": "lag-1",
      "holddown-down-delay": "1000",
      "holddown-up-delay": "5000"
    }
  ]
}
```

**Configuring LAG for Redundancy**

While configuring LAG for redundancy, you must specify the session ID to associate the LAG with the Redundancy Session. LAG can identify its Redundancy session with this mapping.

**Syntax:**

**set link-aggregation interface** <interface-name> **description** <description>
**set link-aggregation interface** <interface-name> **member-interface** <member-interface-name> **lacp-mode** <active | passive>
**set link-aggregation interface** <interface-name> **member-interface** <member-interface-name> **lacp-timeout** <long | short>
**set link-aggregation interface** <interface-name> **member-interface** <member-interface-name> **port-priority**
**set link-aggregation interface** <interface-name> **minimum-link-count** <minimum-link-count>
**set link-aggregation interface** <interface-name> **mode** <static | lacp>
**set link-aggregation interface** <interface-name> **redundancy-session-id** <1-65535>
**set link-aggregation interface** <interface-name> **system-id** <system-id>

| Attribute | Description |
|---|---|
| `description` | Link aggregation interface description. |
| `member-interface-name` | Name of the link member. |
| `minimum-link-count` | Minimum number of active member links required for the link aggregation interface. Default 1. |
| `lacp-mode <active \| passive>` | Mode of the link aggregation interface, static or lacp. Default mode: lacp. |
| `redundancy-session-id` | The value for the redundancy group session ID range from 1 to 65535. |
| `system-id` | Redundancy System ID of link-aggregation interface. |

Example for LAG configuration for redundancy.

The following configuration defines a LAG named 'lag-1' with LACP as the protocol. The minimum link count specifies that at least one physical link must be active for the LAG interface to be considered operational. The redundancy session ID (100) associates this LAG with the configured redundancy session. The system identifier identifies the LACP system and the member interface adds a physical port that participates in the aggregated link.

```
set link-aggregation interface lag-1
set link-aggregation interface lag-1 mode lacp
set link-aggregation interface lag-1 minimum-link-count 1
set link-aggregation interface lag-1 redundancy-session-id 100
set link-aggregation interface lag-1 system-id a8:b5:7e:8f:66:43
set link-aggregation interface lag-1 member-interface ifp-0/1/260
```

Example Configuration:

```
supervisor@rtbrick>cbng2.rtbrick.net:: cfg> show config link-aggregation interface
lag-1
{
    "rtbrick-config:interface": [
      {
        "interface-name": "lag-1",
        "mode": "lacp",
        "minimum-link-count": 1,
        "redundancy-session-id": 100,
        "system-id": "11:22:33:44:55:66",
        "member-interface": [
          {
            "member-interface-name": "ifp-0/0/4"
          }
        ]
      }
    ]
  }
```

API To access the RESTCONF API that corresponds to this CLI, click here.

### 6.5.3. LAG Operational Commands

#### Show Commands

##### Viewing LAG Running Configuration

The following command displays the LAG running configuration on the system.

**Syntax:**

```
show config link-aggregation
```

**Example: LAG Running Configuration**

```
supervisor@rtbrick: op> show config link-aggregation
{
  "rtbrick-config:link-aggregation": {
    "interface": [
      {
        "interface-name": "lag-3",
        "mode": "lacp",
        "minimum-link-count": 2,
        "member-interface": [
          {
            "member-interface-name": "ifp-0/0/4",
            "lacp-mode": "active",
            "lacp-timeout": "long"
          },
          {
            "member-interface-name": "ifp-0/0/5",
            "lacp-mode": "active",
            "lacp-timeout": "long"
          }
        ]
      }
    ]
  }
}
```

#### Viewing LAG Information

The following command displays the LAG information.

**Syntax:**

**show lag** <lag-name> <options>

| Option | Description |
|---|---|
| `<lag-name>` | Displays information for a specific LAG interface |
| `detail` | Displays detailed LAG information |
| `mode <mode>` | Displays information for a LAG mode: `static` or `LACP` |
| `lacp` | Detailed LACP information for the LAG member interface. |

**Example: Viewing LAG Information**

```
supervisor@rtbrick: op> show lag detail
LAG interface name: lag-3
  Status             : Resolved
```

```
  Speed              : -
  Minimum link count : 2
  Mode               : lacp
    Member interface name: ifp-0/0/4
      Member Resolution : lacp
      LACP mode         : active
      Timeout           : long
      Actor system ID   : e8:c5:7a:8e:c1:83
      Actor key         : 3
      Partner system ID : 02:ff:ff:ff:ff:00
      Partner key       : 1
    Member interface name: ifp-0/0/5
      Member Resolution : lacp
      LACP mode         : active
      Timeout           : long
      Actor system ID   : e8:c5:7a:8e:c1:83
      Actor key         : 3
      Partner system ID : 02:ff:ff:ff:ff:00
      Partner key       : 1
```

#### LAG Hierarchy

Using the commands, you can view LAG hierarchy. It provides a structured view of the properties of the specified LAG interface, including its member interfaces and hierarchical relationships.

Syntax:

**show lag hierarchy** <lag-name>

| Option | Description |
|---|---|
| `<lag-name>` | Specifies the name of the LAG. |

Example:

```
supervisor@rtbrick.net: cfg> show lag hierarchy lag-2
lagd:
  LAG interface name: lag-2
    Status: Resolved
    Type: Lag
    Mode: Lacp
    Member interface name: ifp-0/0/12
      Member resolved state: True
    Member interface name: ifp-0/0/13
      Member resolved state: True
    Admin status: Up
    Link status: Up
    Operational status: Up
    Mac address: f4:1e:5e:02:da:fa
```

```
    Minimum link count status: Up
    Minimum link count: 1
    Layer2 mtu: 9000
    Holddown: Enable
    Stability: Holddown
ifmd:
  LAG interface name: lag-2
    Status: Resolved
    Type: Lag
    Mode: Lacp
    Member interface name: ifp-0/0/12
      Member resolved state: True
    Member interface name: ifp-0/0/13
      Member resolved state: True
    Admin status: Up
    Link status: Up
    Operational status: Up
    Mac address: f4:1e:5e:02:da:fa
    Minimum link count status: Up
    Minimum link count: 1
    Layer2 mtu: 9000
    Holddown: Enable
    Stability: Stable
fibd:
  LAG interface name: lag-2
    Status: Resolved
    Type: Lag
    Mode: Lacp
    Member interface name: ifp-0/0/12
      Member resolved state: True
    Member interface name: ifp-0/0/13
      Member resolved state: True
    Admin status: Up
    Link status: Up
    Operational status: Up
    Mac address: f4:1e:5e:02:da:fa
    Minimum link count status: Up
    Minimum link count: 1
    Layer2 mtu: 9000
    Holddown: Enable
    Stability: Holddown
    VPP
      Status: Download Success
      Hw if name: BondEthernet2
      Hw if index: 39
      Sw if index: 46
    BCM
      Ifmapping_pp_port: 85
      Trunk id: 33554434
      Trunk gport: 234881026
```

#### Viewing LAG QoS Policer Counters

The following command displays the QoS Policer Counters.

**Syntax:**

**show qos policer counter**

**Example 1: QoS policer counter**

```
supervisor@rtbrick: op> show qos policer counter
Interface                        Level  Units      Total            Received         Dropped
lag-3                            1      Packets    910              903              7
                                        Bytes      131342           130320           1022
lag-3                            2      Packets    0                0                0
                                        Bytes      0                0                0
lag-3                            3      Packets    0                0                0
                                        Bytes      0                0                0
lag-3                            4      Packets    0                0                0
                                        Bytes      0                0                0
```

**Syntax:**

**show qos policer counter** <lag interface>

**Example 2: QoS policer counter for a specified LAG**

```
supervisor@rtbrick: op> show qos policer counter lag-3
Interface                        Level  Units      Total            Received         Dropped
lag-3                            1      Packets    910              903              7
                                        Bytes      131342           130320           1022
lag-3                            2      Packets    0                0                0
                                        Bytes      0                0                0
lag-3                            3      Packets    0                0                0
                                        Bytes      0                0                0
lag-3                            4      Packets    0                0                0
                                        Bytes      0                0                0
```

# 6.6. HQoS

## 6.6.1. Hierarchical Quality of Service (HQoS) Overview

Hierarchical Quality of Service (HQoS) is a traffic classification and prioritization framework that allows you to provide different Service Level Agreements(SLAs) on bandwidth usage. It allocates network resources to services on a prioritized basis. This is achieved by classifying, policing, shaping, scheduling and remarking the traffic based on service types. The HQoS service can be

applied to Ethernet, IPv4, IPv6 or MPLS packets. The framework supports QoS at multiple levels. At only one level it allows you to allocate resources between services, but when configured at multiple levels through HQoS, it allows complex prioritization schemes.

The RtBrick Full Stack (RBFS) uses the following HQoS mechanisms:

- **Classifier:** Classifies each incoming packet as belonging to a specific class, based on packet contents. Supported classifiers are Behavior Aggregate (BA) and Multifield (MF). In the BA classifier, packets are classified according to the CoS field: IEEE 802.1p, IPv4/v6 ToS/TC, or MPLS EXP. In the MF classifier, packets are classified using additional fields in the IP header: source IPv4/IPv6 prefix, destination IPv4/IPv6 prefix, L4 source port, L4 destination port, and/or IP protocol.
- **Policer:** Policer is implemented in the ingress to drop the unwanted traffic. Policer supports the Committed Information Rate (CIR), the Committed Burst Size (CBS), the Peak Information Rate (PIR), and the Peak Burst Size (PBS). Drop behavior is to either mark traffic as green, yellow, or drop.
- **Queuing:** Drop unqualified packets in advance using the Weighted Random Early Detection (WRED) technology in the case of congestion to ensure bandwidth for qualified services. This is performed at the egress.
- **Scheduler:** Manage traffic on a device using different algorithms for queue scheduling. Such algorithms include Fair Queuing (FQ), Weighted Round Robin (WRR), and Strict Priority (SP). Queues can be connected to schedulers and schedulers can also be connected to other schedulers.
- **Shaper:** Shaper is implemented in the egress, traffic shaping involves buffering and delaying traffic to shape the flows. Shapers are implemented either on queues or schedulers.
- **Remarking:** Remarking allows you to rewrite the outgoing packet's codepoint. Remarking can be performed in the ingress or the egress side of the hardware pipeline.

The figure below shows how QoS does ingress and egress traffic management

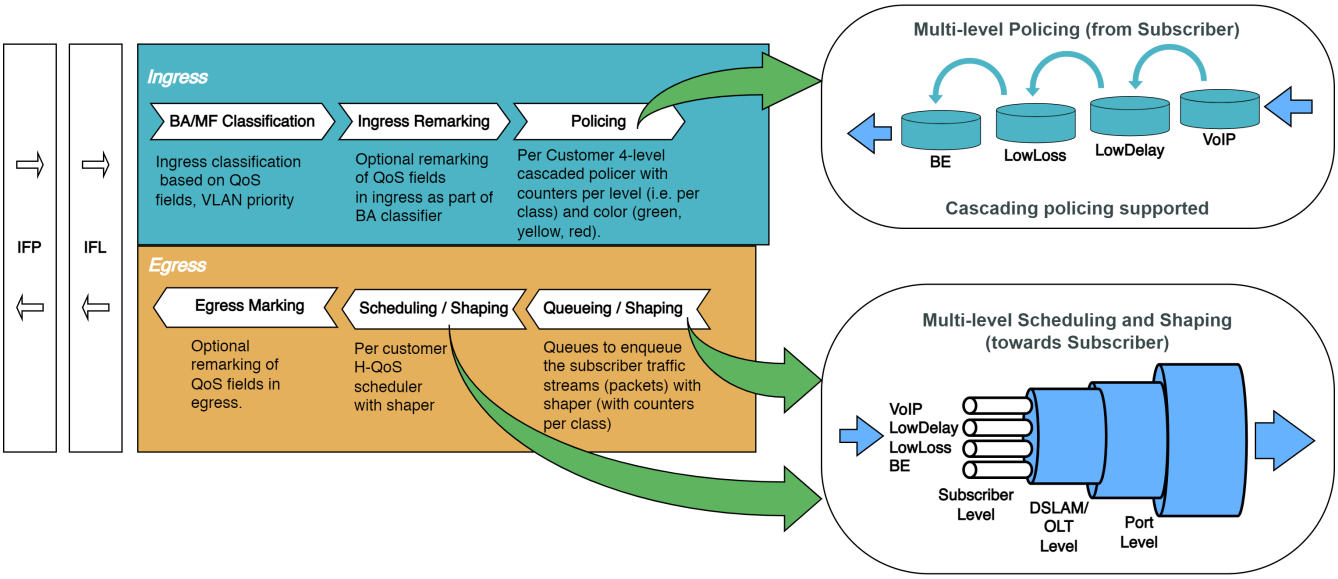



*Figure 118. Ingress and Egress Traffic Management in QoS*

The sections below provide a description of various HQoS components that are configured for both

ingress and egress traffic.

- QoS Profiles
- Behavior Aggregate (BA) Classifier
- Multifield (MF) Classifier
- Remarking
- Policer
- Class-Policer-Map
- Queueing
- Traffic Class to Queue Mapping
- Queue-Group
- Scheduler
- Priority Propagation
- Shaper
- Scheduler Mapping
- L2TP QoS
- Multi-level H-QoS : Level-1 to Level-5

**QoS Profiles**

A profile configuration defines the QoS profile that is attached to either a Subscriber interface or an L3 interface.

Profile maps the following QoS constructs to a Subscriber or an L3 interface:

- Behavioral Aggregate (BA) Classifier
- Multifield (MF) Classifier
- Class Policer Map
- Policer
- Class Queue Map
- Scheduler Map
- Remark Map

**Behavior Aggregate (BA) Classifier**

Classifiers assign the class to which a packet belongs. BA classification is performed on the ingress and maps incoming packet codepoint to a predefined class. BA Classification relies upon markings (that is, codepoint) placed in the headers of incoming packets:

- IEEE 802.1p: Priority - 3 bits
- IPv4: Type of Service byte (ToS) - 8 bits.

- IPv6: Traffic Class (TC) - 8 bits.
- MPLS: Experimental bits (EXP) - 3 bits.

Classifier configuration has the following guidelines and limitations:

- IEEE 802.1p and IPv4/IPv6 classifiers are applied on either Subscriber IFL or L3 IFL by attaching the BA classifier to a profile.
- MPLS Exp classifiers are applied either globally or per instance (to support multiple VPN marking schemes) by attaching the classifier globally or to an instance.
- For IPv4, classification can only be performed based on Type of Service (ToS). Currently, DSCP-based classification is not supported.
- For IPv6, it is possible to classify traffic based on Traffic Classes (TC).
- For EXP classification, RBFS uses the uniform mode to copy MSB 3-bits from 8-bits IPv4-ToS or IPv6-TC to the EXP field at the time of MPLS encapsulation at the remote box. IPv4/IPv6 Classifiers do not classify labeled traffic, hence MPLS Classifier is required for the same.
- Default class for Queue or Policer is *class-0*. If for an incoming packet's *codepoint* there is no class mapping configured under a classifier, the packet will be classified as *class-0*.
- RBFS supports 8 **classes**: *class-0* to *class-7*.

#### Ingress Remarking

Ingress remarking is achieved by configuring the "remark-codepoint" field in the Classifier. Ingress remarking rewrites the IPv4-ToS or IPv6-TC field of the incoming packet at the ingress side with configured remark-codepoint. Note that the ingress remarking is not supported for the BA Classifier with match-type MPLS-EXP.

### Multifield (MF) Classifier

Multifield (MF) classifiers assign the class to which a packet belongs based on multiple fields. Unlike the BA classifier, where only CoS fields are used for classification, the MF classifier additionally uses the following fields:

- **class**: traffic class of the packet (class-0 to class-7) set by prior BA classifier
- **source prefix**: source IPv4 or IPv6 prefix
- **destination prefix**: destination IPv4 or IPv6 prefix
- **protocol**: UDP or TCP
- **source port**: UDP or TCP source port
- **destination port**: UDP or TCP destination port
- **qos markings**: IPv4 TOS or IPv6 TC header value

The actions supported by a Multifield Classifier are:

- **class**: traffic class to be set (class-0 to class-7)
- **Remark codepoint**: remark codepoint for ingress remarking

RBFS treats all the incoming IPv4-TOS or IPv6-TC QoS field values in the incoming packet as untrusted. So a user is required to set `action-remark-codepoint` in the MF Classifier configuration to mark the QoS bits in the IP header of the outgoing packet. If `action-remark-codepoint` is not configured in the MF Classifier, the default value 0 shall be marked in the packet.

The Multifield Classifiers can be bound globally (global.qos.global.config) or via QoS profile (global.qos.profile.config). The global Multifield Classifier applies to all traffic from any instance or interface. The Multifield Classifier assigned via the QoS profile applies only to ingress traffic received on the interface where the profile is bound to it.

The Multifield Classifier is processed after BA classification which allows it to match on selected class from BA classification or to change the assigned class by more granular match conditions. Both classification stages (BA and MF) are optional, they can be combined or used alone, controlled by configuration.

Multifield Classifiers cannot be bound to MPLS core interfaces. Therefore, the downstream traffic (from core to subscriber) should be classified via global Multifield Classifier, while upstream traffic (from subscriber to core) can be classified via Multifield Classifier from the QoS profile, which is instantiated per subscriber with an implicit match on ingress logical interface (InLIF).

- RBFS supports 8 **classes**: *class-0* to *class-7*.
- Per instance MF classifier for MPLS traffic is not supported in RBFS because of hardware limitations.
- The default class for Queue is **class-0**. If an incoming packet has no MF classification configured, it will be classified as *class-0*.
- The default Policer class is determined by VLAN priority. If an incoming packet lacks an MF classification mapping, it will be classified according to the packet's VLAN priority bit.
- Priority 1 is reserved for BA Classifier ACL entries, therefore the recommendation is to use Priority starting from 2 for MF Classifier
- If multiple ACL entries are hit in MF having the same priority, the result is unpredictable. So recommendation is to use different priorities for different ACL entries.

**Traffic Class Remapping Based on Subscriber Queue Profile**

RBFS supports the remapping of downstream traffic classes based on the subscriber QoS profile. This feature helps prevent packet drops when traffic is classified into higher classes (such as class-4 and above) for subscribers with a 4-queue profile. Although, traffic is still counted per class, with 8 class counters available, all traffic from class-4 to 7 is redirected to the queue assigned to class-0. The remapping is automatically enabled for all 4-queue subscribers but is limited to Q2c access-leaf devices.

This feature helps to reduce queue usage and enables more flexible queue profiles, allowing the use of 4-queue subscribers instead of always requiring 8 queues. This feature is enabled, by default.

#### Match MPLS traffic

If MF Classifier is to be applied for MPLS traffic (that is, DOWNSTREAM traffic), match MPLS traffic has to be configured in the MF ACL. If not configured, traffic may or may not match the MF ACL entry in the h/w.

#### Ingress Remarking

Ingress remarking is achieved by configuring the “action remark-codepoint” in the MF Classifier. Ingress remarking rewrites the IPv4-ToS or IPv6-TC field of the incoming packet at the ingress side with configured remark-codepoint.

#### RADIUS Controlled Dynamic MF Classifier

As described in *RBFS RADIUS Services* document dynamic MF Classifier mapping is supported. The dynamic MF Classifier when configured overrides the MF Classifier mapped via QoS profile for the corresponding subscriber but not other subscribers.

For information about configuring the MF Classifier, see Multifield Classifier (MFC) Configuration.

### Layer 2 Access Control Lists (ACL) Priority Bit Egress Classifier

L2 ACL p-bit egress classifier assigns the class to which a packet belongs based on traffic direction (ingress) and outer VLAN priority fields.

- **direction**: Direction of the packet, that is, ingress
- **service-outer-vlan-priority**: Qualifies the packet based on priority bit in the the outer VLAN (IEEE-802.1p)
- **interface-meta-data**: To qualify the packet based on L2 ACL, interface metadata is mandatory (type-1)

The actions supported by a L2 ACL priority bit Egress Classifier are:

- **forward-class**: traffic class to be set on the qualified packets (class-0 to class-7)

- When EVPN-VPWS L2 interfaces are required to create L2 business services, the shared code points have to be corrected at egress to ensure the correct classification towards the subscriber. The egress classification of packets based on L2 ACL is supported.

### Policer

Policer defines the rate at which certain applications can access the hardware resource. So as to rate-limit the traffic from an application, the policer hard-drops the unwanted packets on the ingress side.

In RBFS, policers support the “**two-rate, three-color**” type in a 4-level cascaded mode. This means that each policer level has two rates (CIR and PIR) and three colors (green, yellow and red) with two token buckets as shown below.

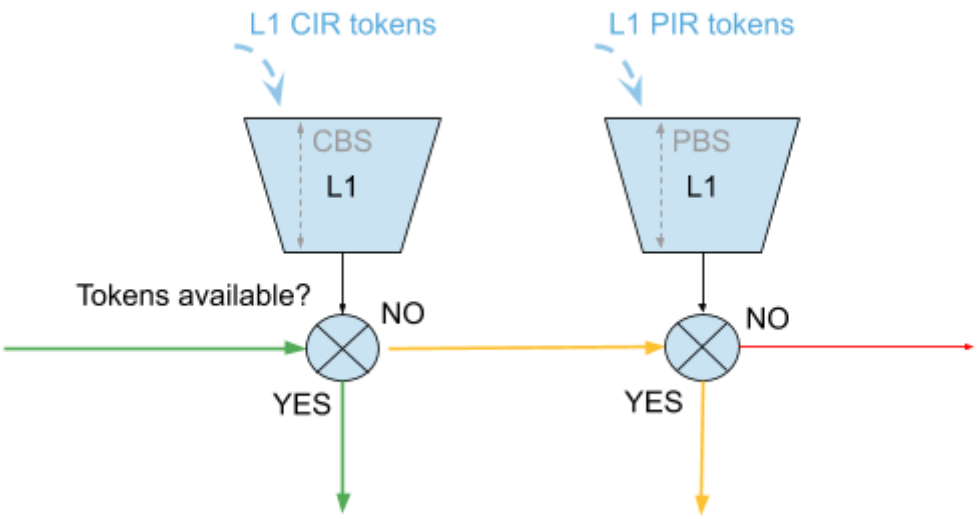



*Figure 119. HQoS Policer Configuration*

This means that traffic below CIR is marked green. Traffic above CIR but below PIR is yellow and above PIR is red. Traffic marked red will be dropped. Traffic marked yellow can be demoted by changing ToS, TC, or EXP using remark map.

In 4-level cascade mode, unused tokens can be passed from higher priority levels to lower priorities where level 1 has the highest and level 4 has the lowest priority as shown in the figure below.

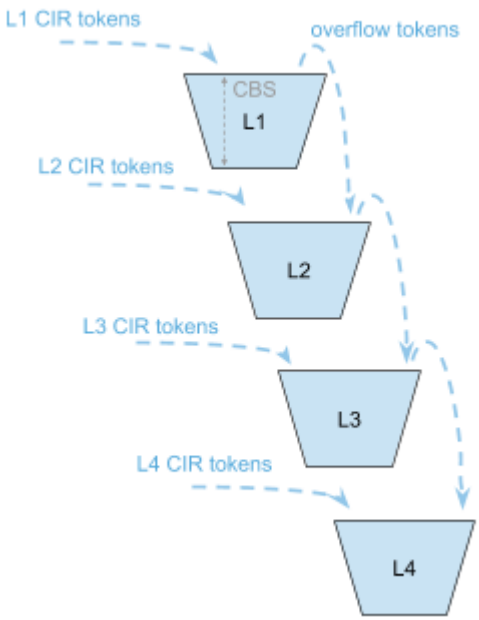



*Figure 120. HQoS Policer Configuration in 4-level Cascade Mode*

Therefore a lower level configured with CIR 0 can still serve traffic if higher priority levels are not consuming all available tokens.

The available tokens per level are calculated by remaining CIR credits from upper levels and additional credits based on configured CIR per level. Per default, the resulting tokens are not limited. The optional max CIR rate attribute limits tokens from CIR and upper levels. Let us assume levels 1 and 2 are both configured with a CIR of 2Mbps. Without max CIR or max PIR (default behaviour) level 2 can reach up to 4Mbps (level 1 CIR/PIR plus level 2 CIR/PIR). This can be limited by max CIR (for example, 3Mbps). If both level 1 and level 2 have a committed information rate

(CIR) of 2Mbps, then level 2 can reach a maximum of 4Mbps (which is the sum of level 1 CIR and level 2 CIR) without any consideration for the maximum CIR. However, the maximum CIR is not relevant for level 1.

In a single-level (levels=1) mode, only level 1 CIR & PIR rates should be configured.

**Example**

| | CIR | RX | TX | PIR | RX | TX |
|---|---|---|---|---|---|---|
| L1 | 2M | 1M | 1M | 2.5M | 1M | 1M |
| L2 | 3M | 20M | 9M | 3.5M / max CIR 8 | 20M | 8M |
| L3 | 4M | 20M | 0M | 4.5M | 20M | 1M |
| L4 | 1M | 20M | 0M | 1.5M | 20M | 0M |
| SUM | 10M | 61M | 10M | 12M | 61M | 10M |

- Here, `M` indicates Mbps (Megabits per second)

In columns 2 through 4 of the preceding example table, L1 consumes only 1Mbps of the available 4Mbps and passes the remaining 3Mbps to L2 which adds 6m based on their own configured CIR resulting in 9m.

In columns 5 through 7 of the preceding example table, L1 consumes only 1Mbps of the available 4Mbps and passes the remaining 3m to L2 which adds 6Mbps based on their own configured CIR resulting in 9Mbps. But because of the CIR limit set to 8Mbps, only 8Mbps of 9Mbps can be used at this level. The remaining 1Mbps is now passed to L3 which does not add additional CIR-based credits. In both examples, L4 would be able to reach up to 10Mbps if upper levels are not consuming credits.

**RADIUS Controlled Dynamic Policer**

The RBFS RADIUS services support dynamic policer rate updates. The dynamic policer rate when configured affects only the QoS instance of the corresponding subscriber but not other subscribers.

**Class-Policer-Map**

Since RBFS supports up to 8 classes but only four policer levels, there is a need to map multiple classes to the same policer level. A *class-policer-map* defines such mappings. Using class-policer-map configuration, one can map any class to any supported policer level (that includes mapping multiple or all classes to the same level). Similar to a policer, a class-policer-map is attached to a profile.

If class-to-level mapping is not configured, no policing will be applied to the traffic for that class.

### Queueing

Queuing helps to drop unwanted traffic in advance at the ingress side in case of congestion. This is to ensure bandwidth for qualified services.

RBFS supports the following queueing techniques:

- Tail Drop (TD): This is a conventional congestion avoidance technique. When the network is congested, drop subsequent packets from the queue.
- Weighted Random Early Detection (WRED): This technique requires configuring “Minimum Threshold”, “Maximum Threshold” and “Drop Probability”, which define the start and end range where packets may get discarded. When the average queue size is below the minimum threshold, no packets will be discarded. The drop_probability parameter can be used to specify the drop probability at the max threshold. When the average queue size is between the min and max threshold, the drop probability increases linearly from zero percent (at the min threshold) to drop_probability percent (at the max threshold). When the average queue size is greater than the maximum threshold, all packets are discarded.
  - When the average queue size is less than the “Minimum Threshold”, no packets will be discarded.
  - When the average queue size is greater than the “Maximum Threshold”, all packets are discarded.
  - When the average queue size is between “Minimum Threshold” and “Maximum Threshold”, the drop probability increases linearly from zero percent (at the minimum threshold) to drop probability (at the max threshold).

- Default queue within a queue group is the one mapped to *class-0*. If the classification is not configured for an incoming packet’s codepoint, the packet will be classified as *class-0*. This will be mapped to queue mapped to *class-0* in *Class-Queue-Map*. For more information, see Traffic Class to Queue Mapping.
- Maximum supported Queue size depends upon DRAM/OCB memory. Since OCB is external memory, hardware does not limit the size that can be configured per Queues.

#### RADIUS Controlled Dynamic Queue

As described for *RBFS RADIUS Services* document dynamic Queue buffer size updates are supported. The dynamic Queue buffer values when configured affect only the QoS instance of the corresponding subscriber but not other subscribers.

#### Traffic Class to Queue Mapping

The traffic class to queue mapping defines the mapping of classes and queues. The Traffic Class to Queue Mapping is attached to a profile.

- You cannot map two classes to the same queue. The class to queue mapping is 1:1.
- If a queue group is created with four queues, only class-0 to class-3 can be

mapped to the queues in class-queue-map; that is, class-4 to class-7 cannot be used.

- In a single-queue/1 queue scenario:
  - All traffic/streams classification [MFC/BA Classifier] will be categorized as class-0 only.
  - Class-0 will be assigned to a single queue.

**Queue-Group**

A Queue Group defines the Queue bundle. A Queue Group contains bundle of either 1 or 4 or 8 queues.

**Scheduler**

A scheduler configuration defines scheduler parameters such as type and shaping rate. The shaping rate defined for a scheduler applies to queue(s) associated with it. NOTE: For 1 Queue in Queue-Group, scheduler is not applicable.

The following scheduler types are supported:

- **Fair Queueing (FQ)**: Uses a round-robin approach to select the next packet to service. This method ensures that all the flows are serviced equally. Configure scheduler type as *fair_queueing* to create FQ scheduler.

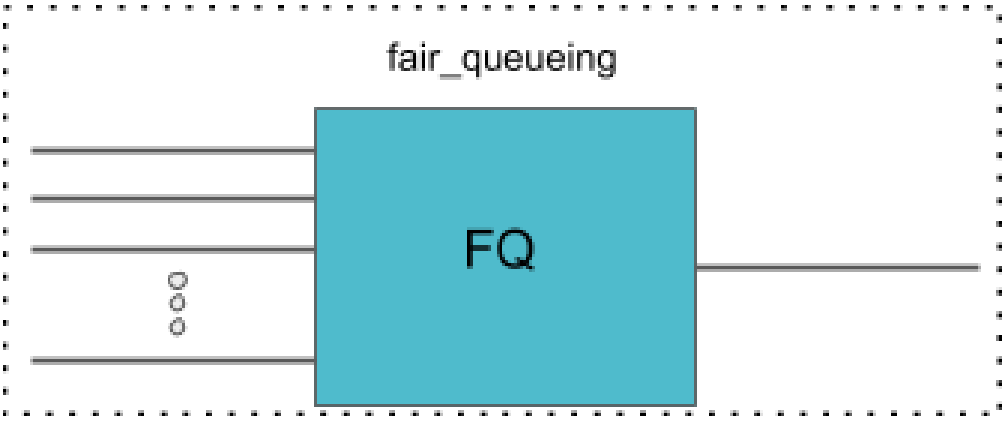



*Figure 121. Fair Queueing (FQ)*

- **Weighted Fair Queueing (WFQ)**: Uses a round-robin approach but with no guarantee of flow being serviced equally (like in FQ). The rotation of the next packet to service is based on the weight that is assigned to each flow. Configure scheduler type as *weighted_fair_queueing* to create WFQ scheduler.
  - Supported weight: 1 to 253

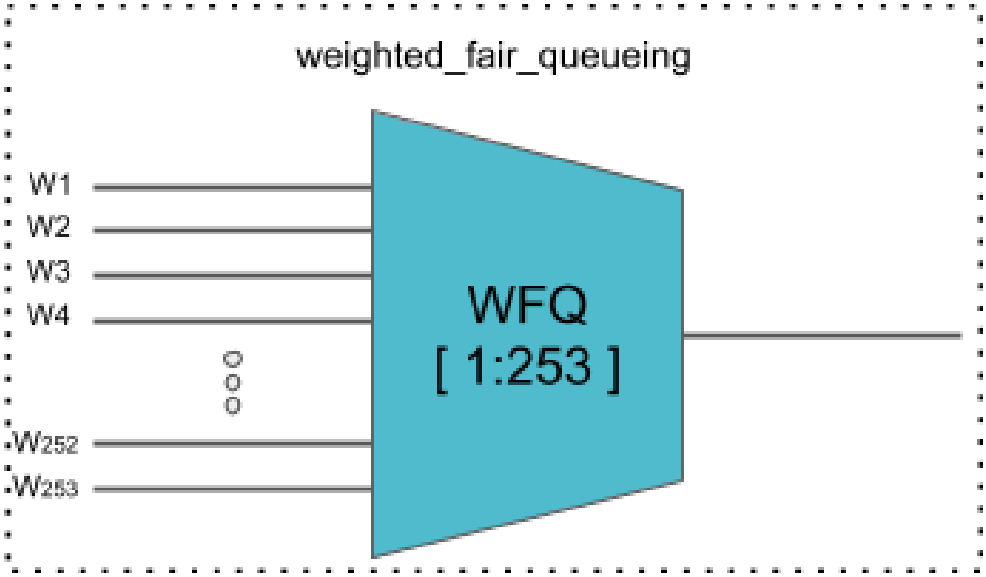


*Figure 122. Weighted Fair Queueing (WFQ)*

> In any WFQ scheduler, the lower the weight, the higher the bandwidth portion is awarded.

- **Strict Priority (SP)**: Uses priority-based approach to service the flow. SP schedulers are supported in “hybrid” mode only. Hybrid mode combines FQ-WFQ schedulers using strict priority.

> The priority order for SP is: **strict_priority_0** > **strict_priority_1** > **strict_priority_2** > **strict_priority_3** (where **strict_priority_0** being highest priority and **strict_priority_3** being lowest)

The following SP scheduler types are supported:

- **2 Strict Priority (2SP)**: Uses SP between 1-FQ and 1-WFQ. There are the following types of 2SP hybrid schedulers:
  - type **“2sp_wfq_independent”**
    - Supported weight: 1 to 63

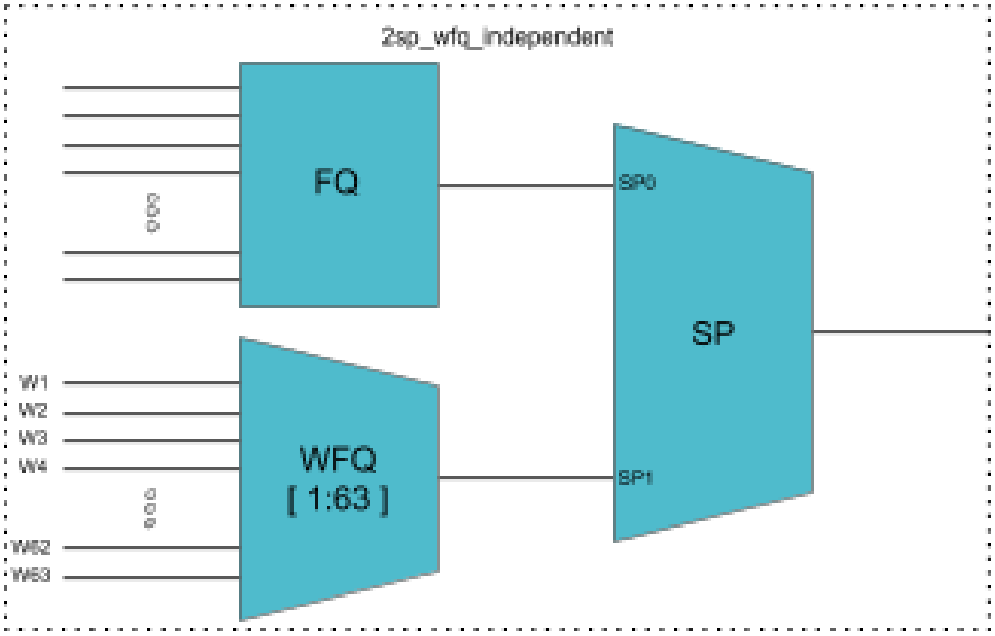



*Figure 123. 2sp_wfq_independent*

- type **“2sp_wfq_discrete”**
  - Supported weight: { 1, 2, 3 }

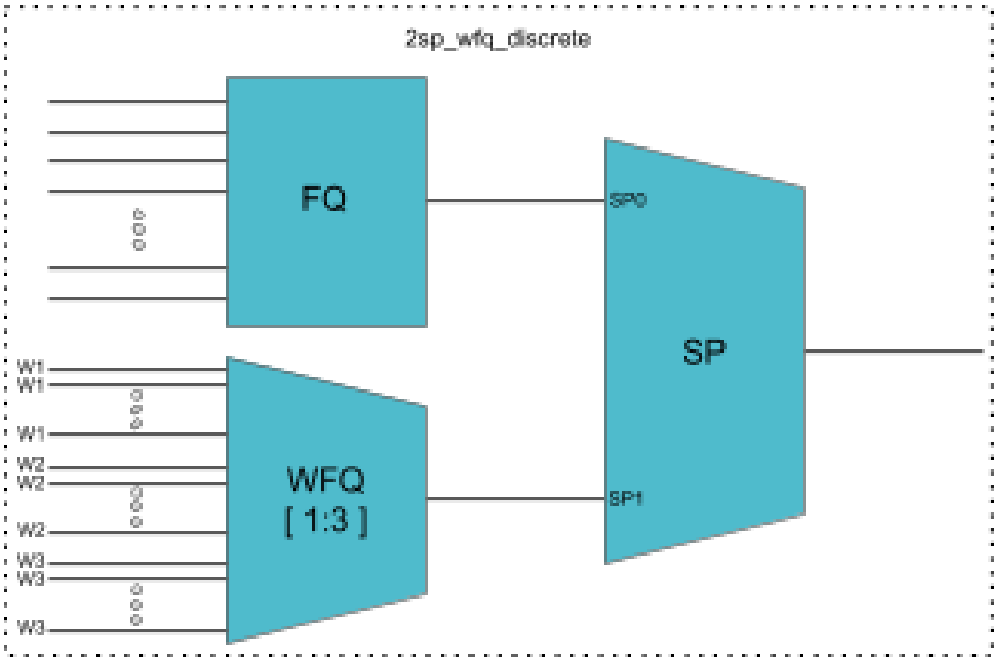



*Figure 124. 2sp_wfq_discrete*

- type **“wfq_independent_2sp”**
  - Supported weight: 1 to 63

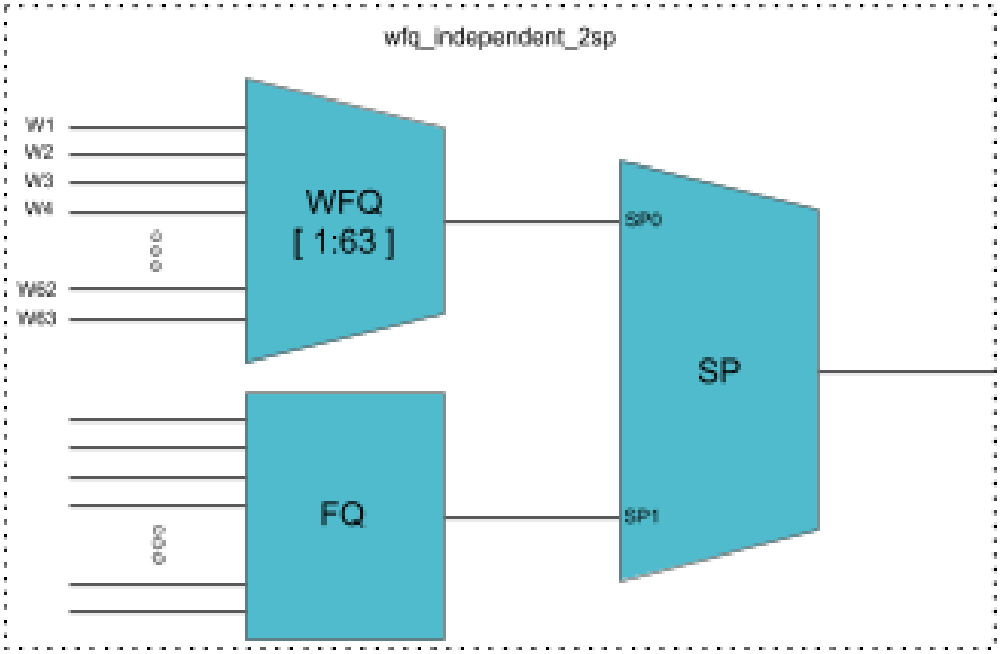



*Figure 125. wfq_independent_2sp*

- type **“wfq_discrete_2sp”**
  - Supported weight: { 1, 2, 3 }

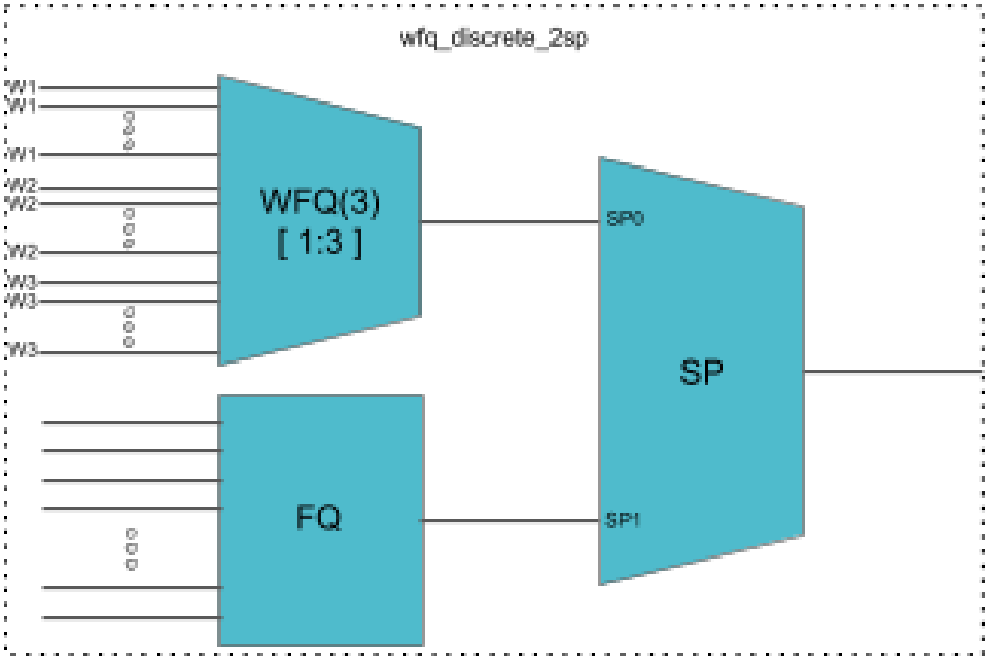



*Figure 126. wfq_discrete_2sp*

- **3 Strict Priority (3SP)**: maps 2-FQs and 1-WFQ
  - type: **“3sp_wfq_discrete“**
    - Supported weight: { 1, 2 }

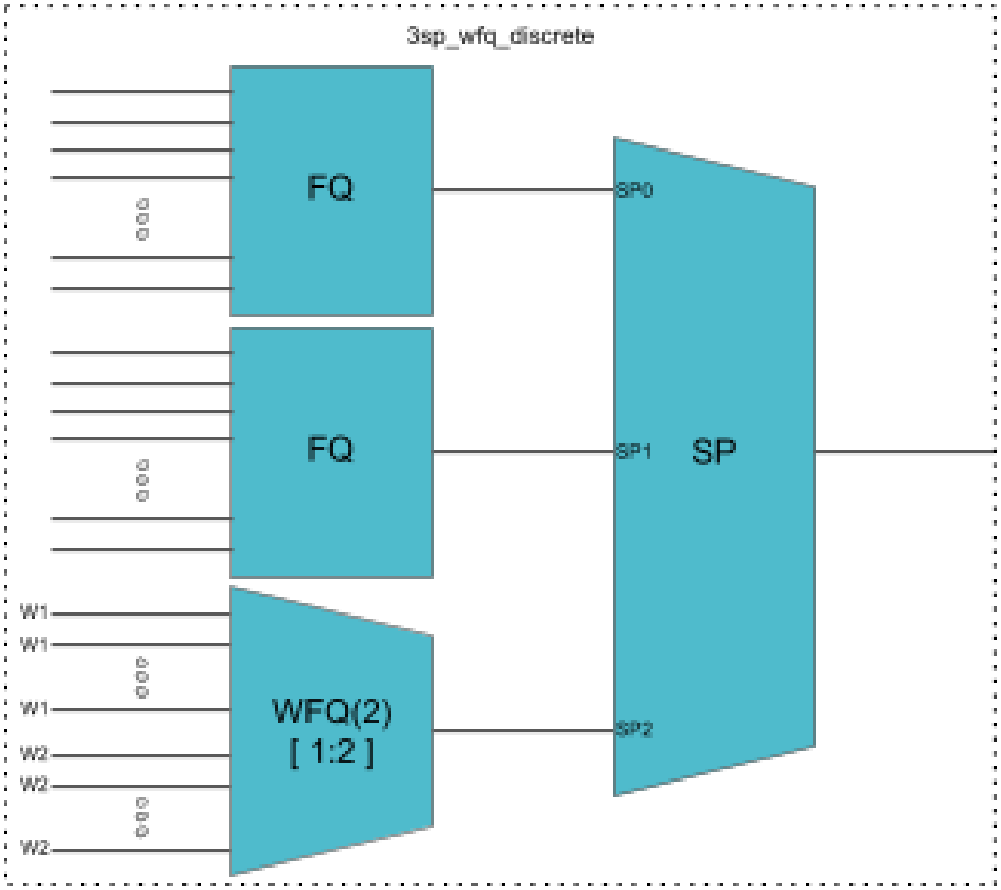



*Figure 127. 3sp_wfq_discrete*

- **4 Strict Priority (4SP)**: maps 4-FQs using SP
  - type **“strict_priority"**

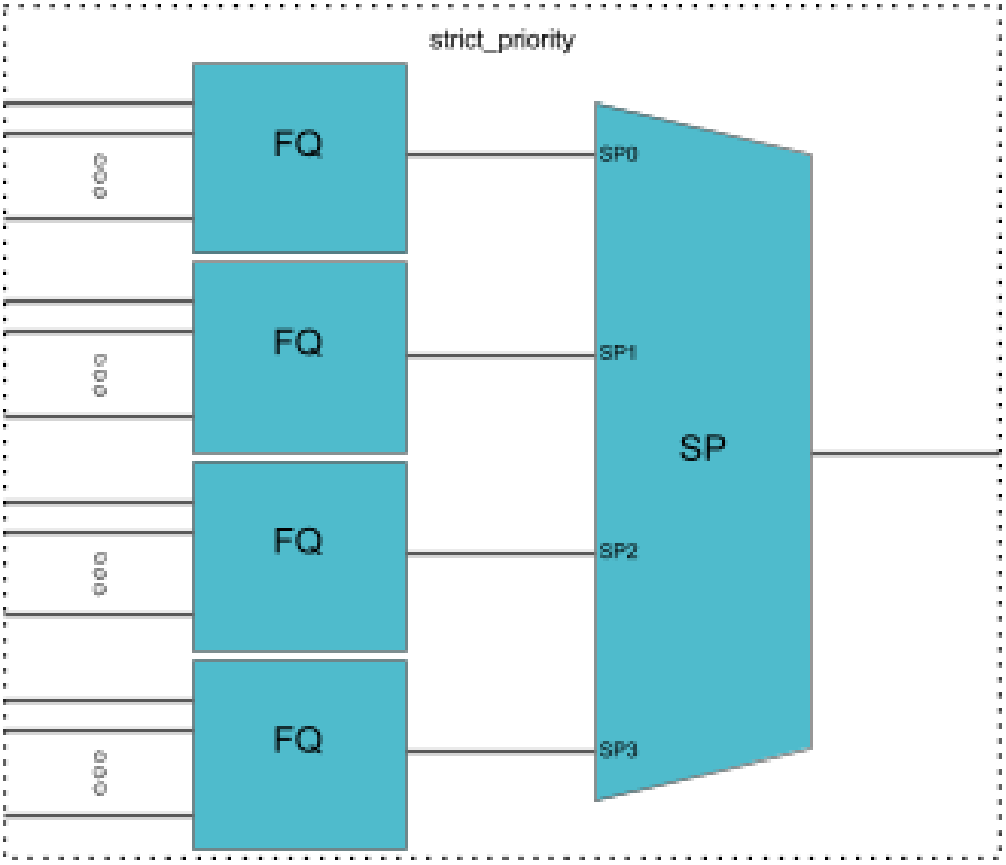



*Figure 128. strict_priority*

**Priority Propagation**

Hierarchical QOS (HQoS) on RBFS is implemented by connecting or chaining queues to scheduler elements (Q —> SE), scheduler elements to each other (SE —> SE), and scheduler elements to ports (SE —> PORT). Each scheduler element can have different child connection points based on types described in the section Scheduler.

This means that sched_0 in the example below is not scheduling between the attached queues, but between the different child connection points SP0 to SP3. The scheduler element sched_0 cannot differentiate between Q1 and Q2 in this example because both are connected to SP2.

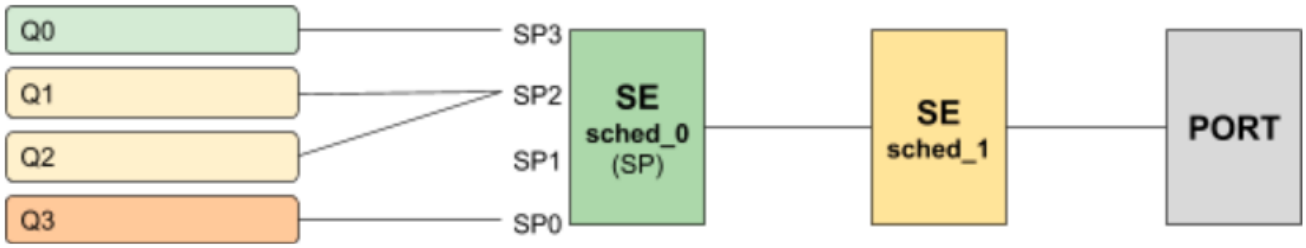



*Figure 129. Priority Propagation*

Without priority propagation, each scheduler element can have multiple child connection points but just one parent connection point. Therefore traffic leaving a scheduler element cannot be differentiated by the parent scheduling element. The parent scheduler element sched_1 receives the traffic from sched_0 on the selected child connection point. As already mentioned scheduling within a scheduler element happens between child connection points. Second, a scheduler element has only one parent connection point which can be connected to a child connection point of another scheduler element (output of sched_0 → input of sched_1). This results in the situation that all traffic from this SE is handled equally regardless of the queue. This may lead to dropped priority traffic like voice or control traffic in case of congestion in parent elements. For example, if sched_1 has a shaping rate lower than the one of sched_0, it will drop traffic unaware of its original priority.

This problem is addressed with priority propagation which is enabled by default.

With priority propagation, the scheduler elements operate in a dual-flow mode with high and low-priority flows. The credits generated from the physical interface will be consumed by all attached high-priority flows first and only the remaining credits will be available for low-priority flows. In this mode, an implicit FQ element is created for each scheduler element. All queues assigned to low-priority flow will be attached to this element.

An additional composite option of the scheduler element allows also the differentiation between multiple low-priority queues if required. This composite type is created implicitly and does not need to be configured.

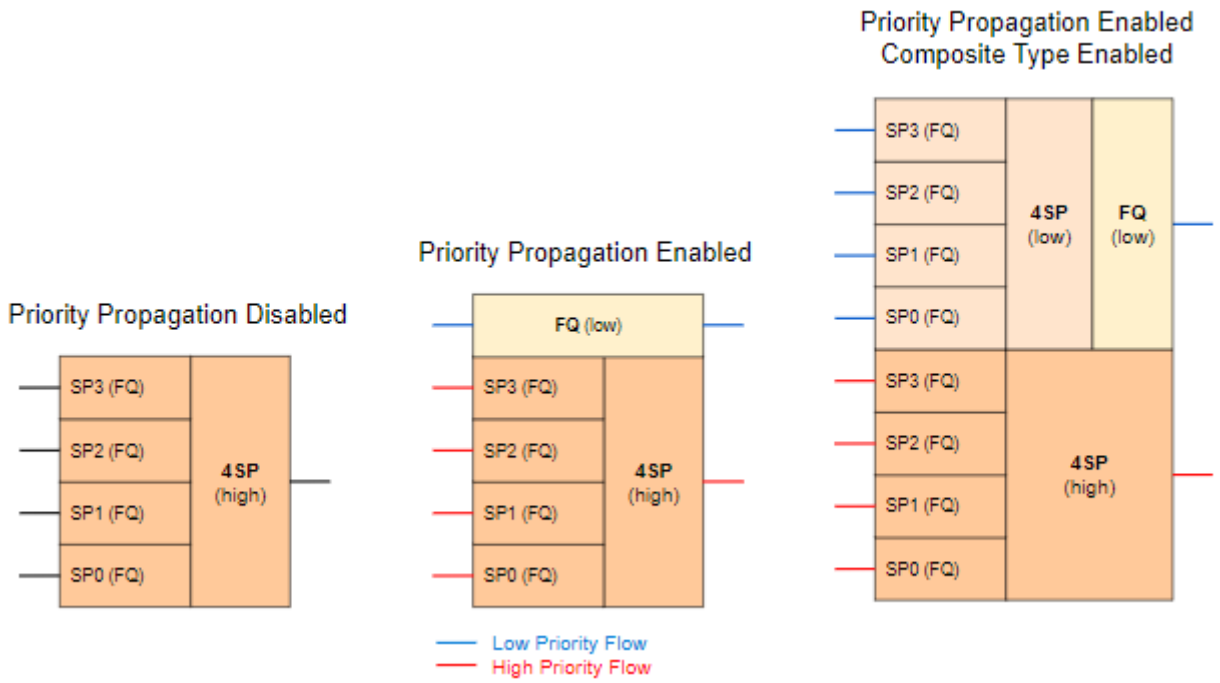



*Figure 130. Priority Propagation - Composite Option*

Without priority propagation enabled, each scheduler element consumes only one scheduler resource compared to two elements if enabled. The composite type consumes three scheduler elements.

With priority propagation disabled, all traffic is considered a high-priority flow.

Now for each queue, we can select if connected to high-priority or low-priority flow where high-priority flow is selected per default if not explicitly mentioned.

Assuming the example as before but with priority propagation and Q0 assigned to low-priority flow and Q1 - Q3 assigned to high-priority flow.

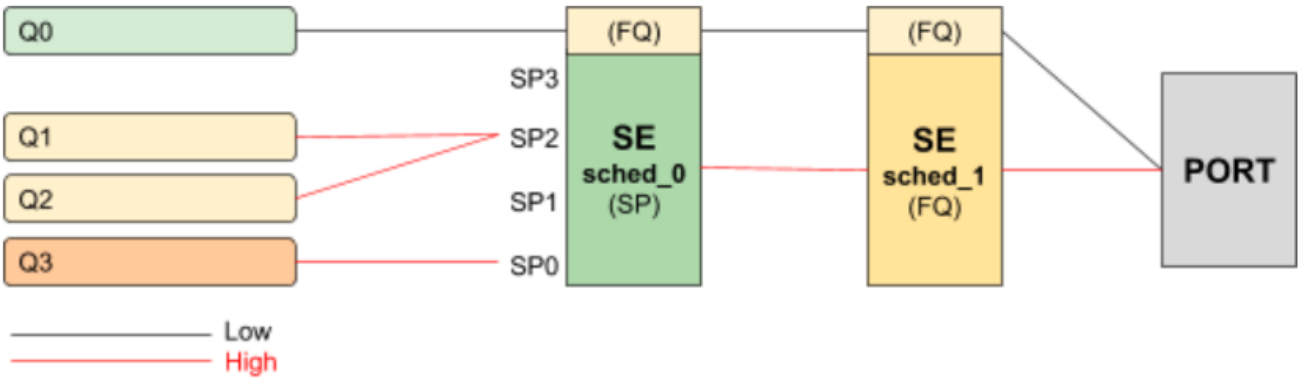



*Figure 131. Priority Propagation - Low/High Priority Flows*

The figure below shows a typical multi-level QoS configuration without priority propagation on the left and with priority propagation on the right side.

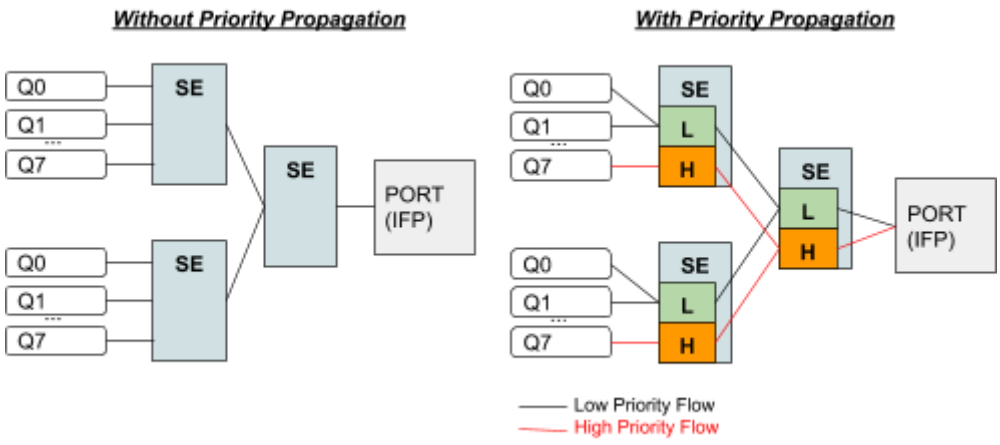



*Figure 132. Multi-level QoS configuration with/without priority propagation*

The credits generated from the physical interface will be consumed by high-priority flow first and the remaining credits will be available for low-priority flow. The high-flow traffic at any one element is scheduled based on the type and connection point. Between schedulers, it depends on how they are connected to the parent scheduling element. Per default all levels there is FQ for low and FQ for high-priority flows. The port scheduler is also FQ.

In this mode, each shaper supports two different rates for low and high-priority where the actual shaper rate is the sum of low and high-priority rates. If the low-priority rate is zero, this flow is only served if the high-priority flow is not consuming all credits. An example might be a high rate of 9Mbps and a low rate of 1Mbps which results in 10Mbps for low-priority flow if high-priority flow is not consuming any packets but at least 1Mbps is ensured.

The following example shows a typical access service provider configuration with priority propagation enabled with and without composite type.

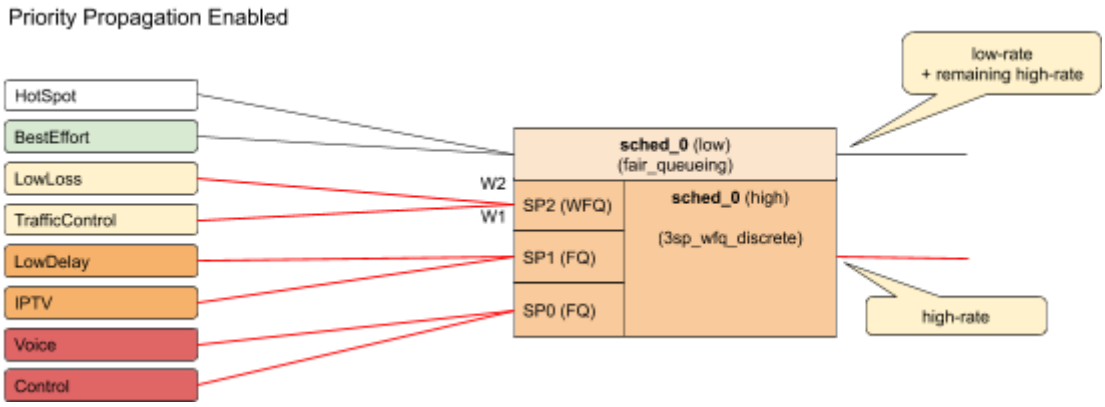



*Figure 133. Priority Propagation Enabled*

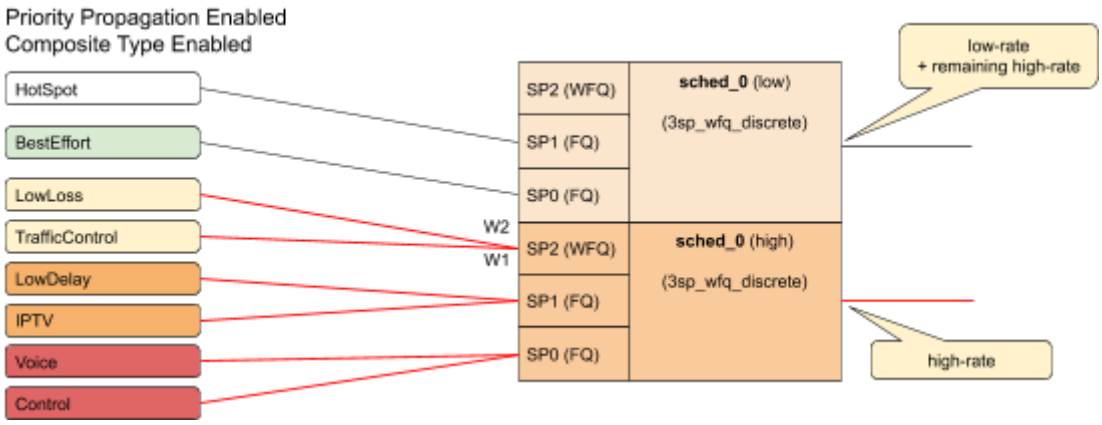



*Figure 134. Priority Propagation and Composite Type Enabled*

**Simple Priority Propagation Scheduling Example**

Without priority propagation, the parent scheduler drops traffic equally from all classes as it is unaware of priorities:

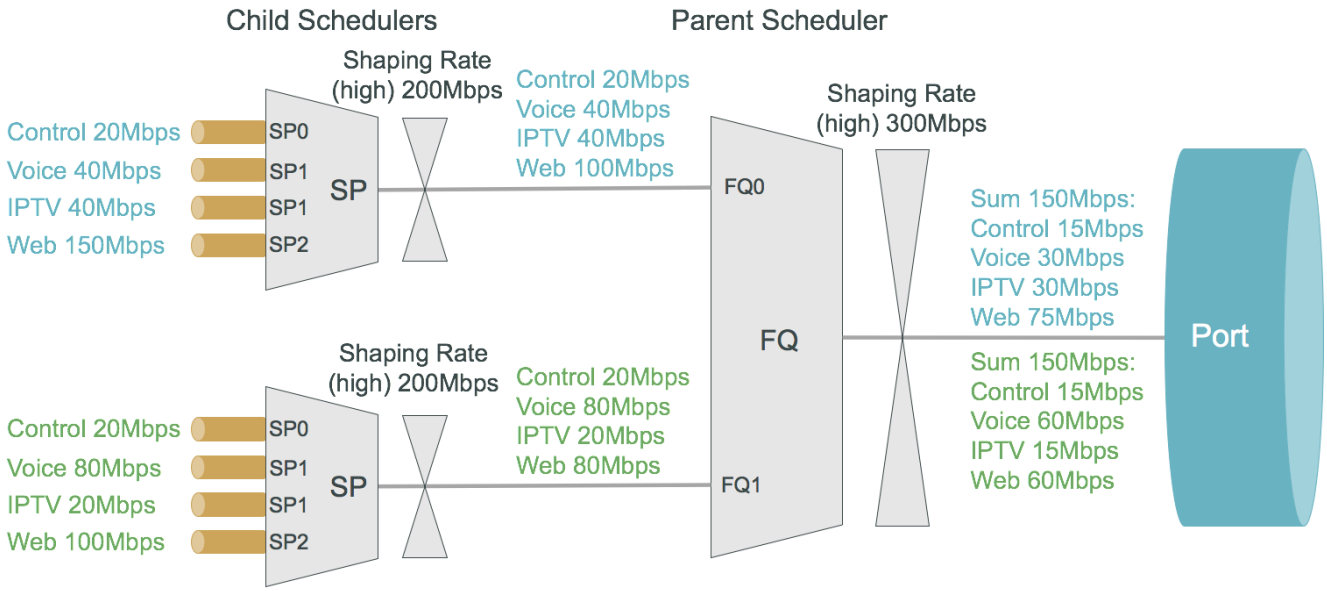



*Figure 135. Priority Propagation Scheduling Example*

With priority propagation, the parent scheduler serves high-priority flows first as shown in the figure below:

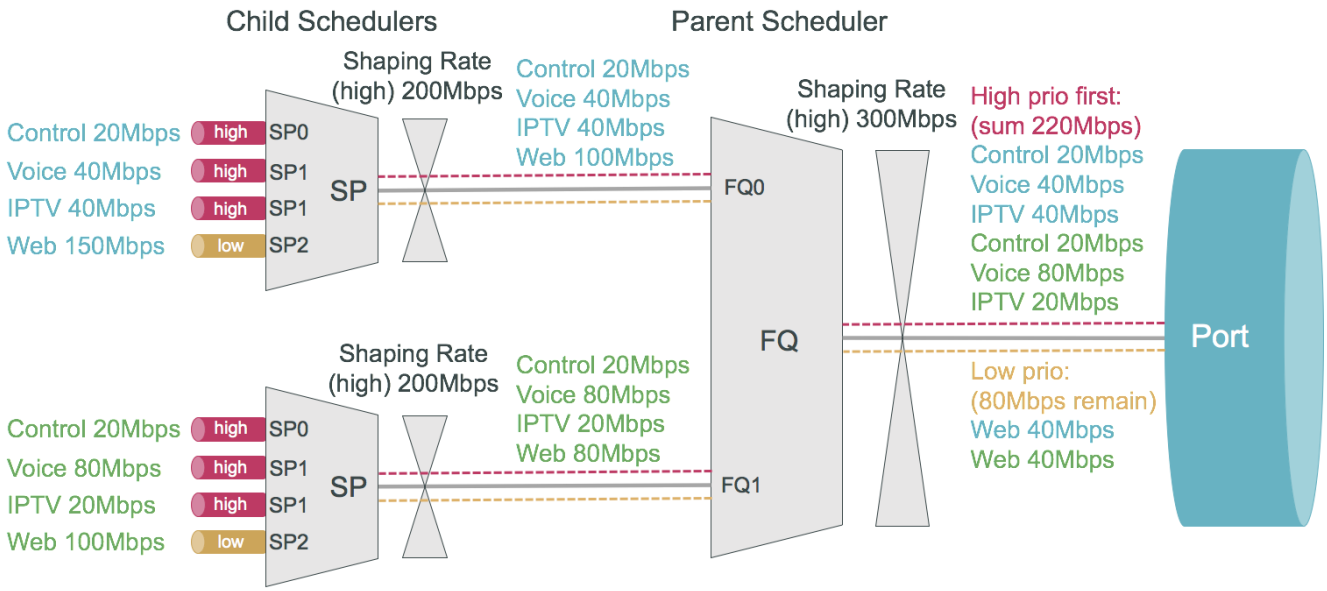



*Figure 136. Priority Propagation Scheduling Example*

With priority propagation and dual-flow shaping, the parent scheduler serves high-priority flows first up to the high-flow shaping rate:

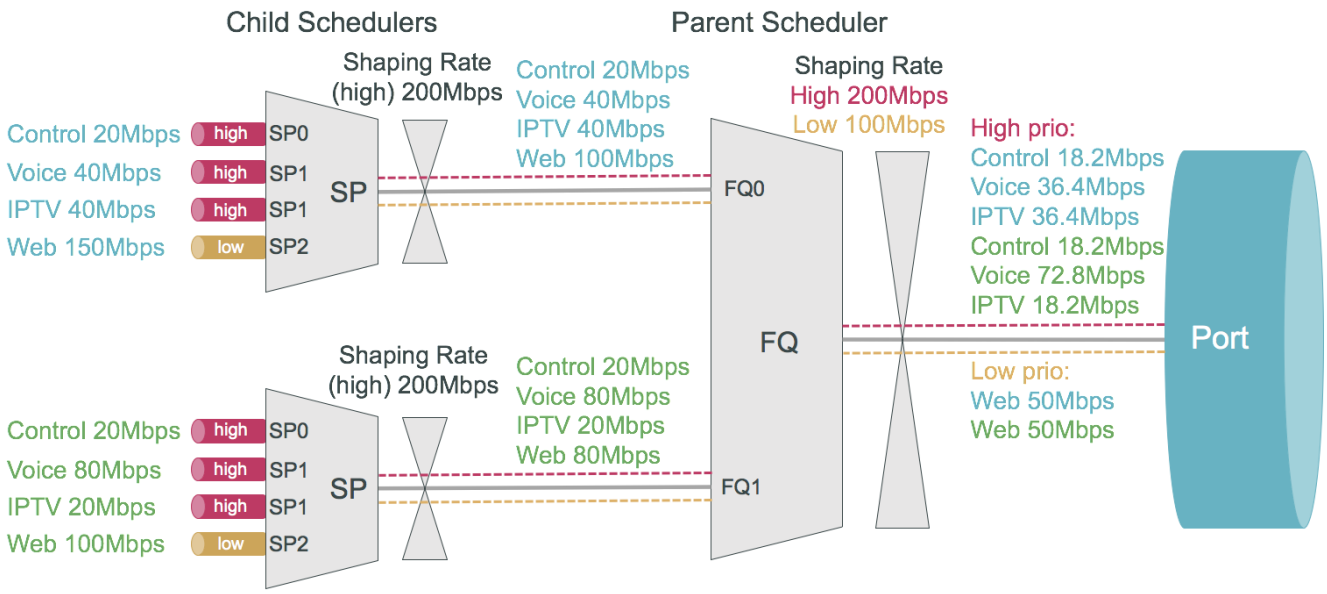



*Figure 137. Priority Propagation Scheduling Example*

### Shaper

Shaper is used to rate-limit the traffic at the egress. In RBFS, shapers can be attached to both Queue and Scheduler.

A shaper configuration defines the shaping rate in Kilo-bits-per-seconds (Kbps).

- Setting the shaping rate to 0 (zero) sets the rate to unlimited. Hence it is recommended to configure at least 1 Kbps so that shaping takes place.
- If shaping rates are to be unconfigured, it needs to be done at all scheduler levels.

#### Low-rate Shaping

The Low-rate Shaping feature performs queueing and scheduler-level traffic shaping to rates lesser than 1000 Kbps so that the higher-priority (voice) traffic to flow at optimal levels.

> Low-rate Shaping is supported only on high-priority flows, that is high-flow configuration parameter.

RBFS Access-Leaf and Multiservice Edge platforms have been enabled with Low-rate Shaping by default. For information about the Low-rate Shaping feature enabled platforms, see section 'Feature/Resource Usage" in the *Platform Guide*.

#### RADIUS Controlled Dynamic Shapers

RBFS RADIUS services support dynamic shaper updates. The dynamic shaper when configured affects only the QoS instance of the corresponding subscriber but not other subscribers.

#### Scheduler Mapping

Scheduler Map defines the set of relationships between parents and children in egress scheduling hierarchy. A child in a Scheduler Map configuration can be either Queue or Scheduler. A parent in a Scheduler Map configuration can be either a Port or Scheduler.

> For 1 Queue in Queue-Group, scheduler-map is not applicable. **Connection Point and Weight**

Child-queue or child-scheduler in a scheduler map configuration is connected to the parent-scheduler at “**connection point (CP)**“. Connection point configuration also has “**weight**” associated with it if the parent has a WFQ scheduler corresponding to that connection point. The valid connection point value for a child to connect to parent **WFQ/FQ** scheduler is **no_priority** and to connect to parent **SP/Hybrid** scheduler is between **strict_priority_0** to **strict_priority_3** (based on a number of Strict Priority points in parent scheduler).

#### Connection Types

There are five connection types in a scheduler map entry:

- queue_to_port
- queue_to_scheduler
- scheduler_to_scheduler
- scheduler_map_to_scheduler_map
- scheduler_to_port

> - For the **queue_to_port** connection type, the scheduler has no role.

#### Remarking

The packet markers set the codepoint in a packet to a particular value, adding the marked packet to

a particular behavior aggregate. When the marker changes the codepoint in a packet, it "remarks" the packet. The codepoint in a packet can be IPv4-ToS, IPv6-TC, MPLS-EXP, or IEEE 802.1p field.

The following remarking options are supported in RBFS:

- IEEE 802.1p : Priority - 3 bits.
- IPv4: Type of Service byte (ToS) - 8 bits.
- IPv6: Traffic Class (TC) - 8 bits.
- MPLS-IPv4: MPLS Experimental bits (EXP) - 3 bits.
- MPLS-IPv6: MPLS Experimental bits (EXP) - 3 bits.

IPv4/v6 and IEEE 802.1p remark-map are applied on an interface - subscriber-ifl or l3ifl using Profile Name.

MPLS-IPv4/v6 remark-map is applied either globally or per instance (to support multiple VPN marking schemes) using Remark-Map Name.

In RBFS, remarking can be performed at the ingress or egress:

- **Ingress remarking** is achieved by configuring the **remark-codepoint** field in the Classifier. Ingress remarking rewrites the IPv4-ToS or IPv6-TC at the ingress side with configured remark-codepoint. The configured remark-codepoint can be modified again at the egress side using remark-map. The ingress remarking is supported for IPv4, IPv6, and IEEE 802.1p BA classifiers. Ingress remarking is supported in MF Classifier as well.
- **Egress remarking** is achieved by configuring the **remark-map**. Remark Map is the mapping of **match-codepoint** and **color** to **remark-codepoint**. Egress remarking helps to remark the IPv4-ToS / IPv6-TC field in the IP header, or to write the EXP field in the MPLS label(s), or to write the IEEE 802.1p field in the VLAN header.

Here *Color* is used to set different *remark-codepoint* for same *match-codepoint* based on color marked by the Policer (i.e. *green* or *yellow*). Color is a mandatory field in remark-map. To set the same *remark-codepoint* for a *match-codepoint* irrespective of color, we have to set color as “*all*”.

**IPv4-ToS, IPv6-TC, or MPLS-EXP remarking:**

- If the *remark-codepoint* is not configured in the BA Classifier or there is no hit in MF Classifier, match-codepoint in the remark-map is the ToS/TC value of the incoming IP packet.
- If the *remark-codepoint* is configured in the BA Classifier and there is no hit in the MF Classifier, match-codepoint is the same value as the remark-codepoint in the BA Classifier
- Irrespective of the *remark-codepoint* configured in the BA Classifier, if there is a hit in the MF Classifier the *match-codepoint* is the same value as the action remark-codepoint (0 if no action *remark-codepoint* configured) in the MF Classifier.

**IEEE 802.1p VLAN remarking:**

- Class-to-IP based Remark Map for L2TP UPSTREAM traffic is mapped globally. For more information, see the L2TP QoS section.

- In tunnel termination cases (i.e. Downstream traffic from core to Subscriber) the *remark-codepoint* in the MPLS BA Classifier is of no use. Therefore the *match_codepoint* in remark-map at the egress shall be the ToS/TC value of the incoming IP packet.
- In IP tunnel encapsulation cases (i.e. L2TP Upstream traffic from Subscriber to core) the remark-codepoint in the IPv4-TOS BA Classifier is of no use. Therefore the match_codepoint in class-to-ip remark-map at the egress shall be the Class derived from ingress BA Classifier.
- If no MPLS remarking is configured for the Upstream traffic, EXP bits in the MPLS header are derived from IP header TOS/TC bits using the Uniform MPLS mode.
- For VLAN: Only class to IEEE 802.1p remarking is supported.
- For IPv4: Only ToS based remarking is possible. DSCP-based remarking is not possible.
- For IPv6: Only TC based remarking is possible.
- The VLAN priority remarking support for the platforms is as follows:
  - **UfiSpace S9600-72XC**:
    - match-codepoint: Value of the IPv4-ToS/IPv6-TC in the incoming IP packet or the remark-codepoint configured in the BA/MF Classifier
    - remark-codepoint: Based on the VLAN priority
  - **Edgecore AS5916-54XKS**:
    - match-codepoint: MF or BA classifiers are used to determine class at the ingress
    - remark-codepoint: Based on the VLAN priority
  - **QAX Platform: UfiSpace S9500-22XST**
    - In QAX platform, IP TOS to VLAN bits do not get directly mapped. The IP ToS is mapped to internal priority in classification, and then from internal priority VLAN priority bit remarking is performed.
- **IPv4/v6 and IEEE 802.1p** remark-map is applied on an interface - subscriber-ifl or l3ifl using Profile Name.
- **MPLS-IPv4/v6** remark-map is applied either globally or per-instance (to support multiple VPN marking schemes) using Remark-Map Name.

**L2BSA L2X VLAN Priority Remarking:**

The L2BSA L2X VLAN priority remarking support for the platforms is as follows: Only A10NSP switches support VLAN operations, access-leaf is transparent.

- Downstream Traffic (Remote ISP to Subscriber):
  - On A10NSP Switch:
    - The class derived from the VLAN Pbit Classifier is used to remark MPLS Exp and Outer

VLAN IEEE-802.1 Pbit Value, inner VLAN Pbit is transparent.

  - On Q2C Access Leaf:
    - The received Outer VLAN IEEE-802.1 Pbit value is retained, explicit remarking at access-leaf is not supported.

- Upstream Traffic (Subscriber to Remote ISP):
  - On Q2C Access Leaf:
    - To remark the outgoing packet, global MPLS Exp remark map can be used.
  - On A10NSP Switch:
    - To remark the outgoing packet, VLAN IEEE-802.1 Pbit remark map can be used.

For more information about L2BSA configurations, see the L2BSA User Guide.

**L2VPN VLAN Manipulation QoS Remarking:**

- VLAN remarking allows for various VLAN manipulations (push, swap, push-push, swap-swap, swap-push) on the outer VLAN in double-tag or single-tag scenarios.
- VLAN remarking for VLAN manipulation involving POP operations (pop, pop-pop, pop-swap) is not supported.

### Header Compensation

#### Queue Compensation

The rate at which the packets are dequeued from a queue depends on the credit received by that queue. The source of the credit received by a queue is the egress port to which the queue is mapped. When a packet is dequeued, the credit balance is decreased by the packet size. But, the packet size that is used must be adjusted to model the packet size at the egress, rather than its actual size at the ingress queue. Thus the header compensation is used to adjust for the differences in header size between ingress queue and egress port. RBFS supports static header compensation configuration per queue (in bytes).

#### Port Compensation

Similar to queue header compensation where header compensation is performed at the per-queue level, RBFS supports the following header compensation at the per-port level:

- **Ingress Header Compensation**: In line with the header compensation option that we have per-queue, RBFS supports static header compensation configuration at the ingress to be used by the policing. Header compensation changes the effective size of the packet to compensate for changes in header size (such as the CRC removal) when considering the packet for policing. Unlike queue, RBFS ingress header compensation configuration is per ingress port (in bytes).
- **Egress Header Compensation**: In line with the header compensation option that we have per-queue or per-port at the ingress, RBFS supports static header compensation configuration at the egress. The egress header compensation configuration is per egress port (in bytes).

The supported range for header compensation is -64 to +64 bytes.

**L2TP QoS**

The Layer 2 Tunneling Protocol (L2TP) QoS for upstream is similar to any other locally terminated subscriber. The QoS Profile is mapped dynamically via RADIUS for the L2TP subscribers.

The L2TP QoS for Downstream requires IPv4-TOS based BA Classifier which is mapped to L2TP Tunnel. The same can be achieved by attaching *l2tp-classifier-name* in *global QoS* configuration.

```
forwarding-options {
    class-of-service {
        global {
            l2tp-classifier-name l2tp-ip;
        }
    }
}
```

For Downstream queueing, there is no change. Queueing is applied using QoS Profile similar to locally terminated Subscribers.

The following features are supported for L2TP QoS.

- Upstream
  - BA Classifier: IEEE 802.1p
  - Policing
    - Policer statistics
  - Remarking

> - L2TP upstream traffic can be remarked by configuring remark-code-point in the classifier configuration.
> - L2TP downstream traffic can be remarked by profile with the remark configuration. A match-codepoint value will be the class value derived from the l2tp-ip classifier.

- Downstream
  - BA Classifier: IPv4-TOS
  - Queueing/Scheduling/Shaping
    - Queue statistics
  - Remark-Map: IEEE 802.1p (Class to VLAN priority remarking)

**Guidelines**

- To avoid control traffic policing/shaping, the assumption is that the IEEE 802.1p bits in Upstream or IPv4-TOS bits in Downstream will be different for control and data traffic, control traffic is expected to have the highest precedence.
  - Upstream classification is based on IEEE 802.1p bits.

    - Downstream classification is based on IPv4-TOS bits of the outer IP header.
- PBIT (IEEE-802.1p) classification and policing for the upstream traffic from IPoE subscribers is not supported.
- Mark control plane traffic with codepoints based on protocols and layers (such as routing, signaling, and subscriber-management protocols, layer-2, layer-3) and ensure it is treated with higher priority.
- It is recommended to configure MFC (multi-field classifier) for the upstream classification for subscriber traffic.

**Multi-level H-QoS : Level-1 to Level-5**

A sample 5-level requirement looks like this.

**Level-1 (IFP)**

Physical Interface Shaper.

**Level-2 (PON TREE)**

Each PON tree is a TDM based shared medium with typically ~2.5 GBit/s (GPON) shared by up to 32 consumers (ONT or DPU).

**Level-3 (DPU)**

In case of FTTB there is a single DPU with multiple consumers via G.Fast DSL connected which requires an additional hierarchy. This level is not needed for FTTH or FTTC.

**Level-4 (ANP or Session)**

The Access Node Port (ANP) or outer VLAN level describes a single customer line. This might be an ONT in case of FTTH or DSL interface behind a DPU in case of FTTB. This level can be also represented on PPPoE sessions as long as just one session is permitted per VLAN.

**Level-5 (QUEUE)**

The Queue level shaper is required to limit the class-of-service bandwidth like Voice or IPTV traffic limit.

The figure below shows the diagram along with QoS representing Level-1 to Level-5 Hierarchical scheduling.

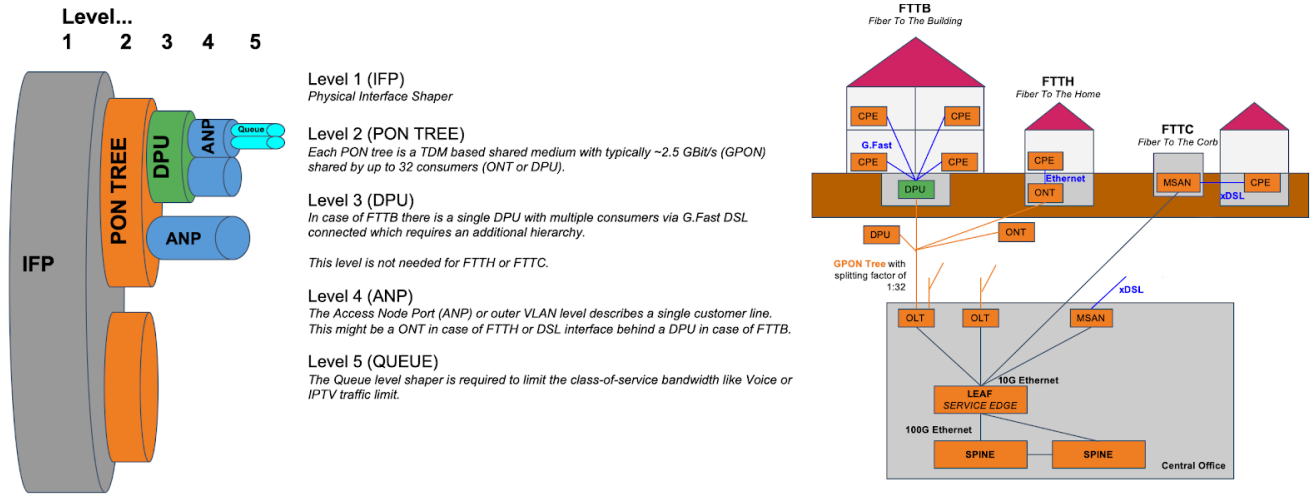



*Figure 138. Level-1 to Level-5 Hierarchical Scheduling*

Levels 4 and 5 are configured per logical interface (i.e. subscriber-ifl or l3-ifl). Separate scheduler-map representing levels 1 to 3 connectivity shall be statically configured and mapped to the corresponding physical interface (IFP).

The child scheduler in a subscriber scheduler map is connected to the parent scheduler in a physical interface scheduler map using the following way:

- Dynamically via RADIUS in case of dynamic subscribers like PPPoE sessions (Subscriber-IFL).

The figure below shows the same details as the preceding figure before with the different levels but from the DPU-PON-IFP scheduler-map point of view.

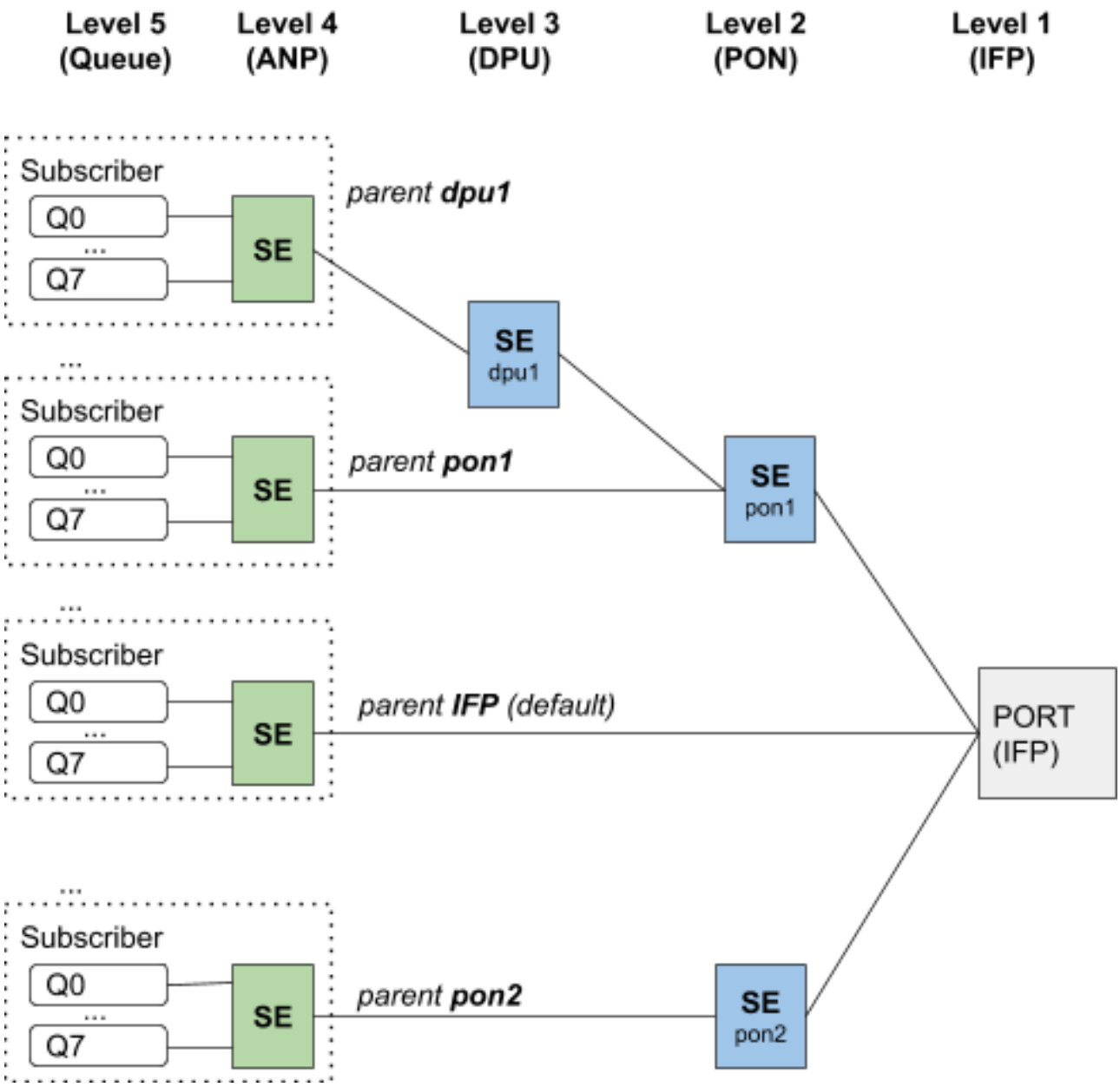



*Figure 139. Level-1 to Level-5 Hierarchical scheduling (DPU-PON-IFP scheduler-map)*

#### QoS Support for L2 and L3 Logical Interfaces

QoS is supported on Layer 2 and Layer 3 logical interfaces.

#### Support Interface-set for Grouping Logical Interfaces

An interface-set is a collection or group of logical interfaces (that is, L2 or L3) belonging to the same physical interface (IFP).

#### QoS Support for Interface-set Hierarchy

QoS can be configured at the interface-set hierarchy (service-group), and the IFLs belonging to this interface-set must configure the service-group scheduler to connect to the PON scheduler or directly to the IFP.

#### Business Service QoS

A business service can be leveraged using an interface-set. A business service comprises of a number of logical interfaces. QoS can be configured at the interface-set hierarchy (service-group) and at the logical interface level as well.

The logical-interface-scheduler is connected to an interface-set (business service group) scheduler and the interface-set-scheduler can be connected to a PON-scheduler or directly to a port-scheduler.

#### MPLS HQoS

The MPLS HQoS has both UNIFORM and PIPE modes. These modes provide the following functionality:

- During MPLS Encapsulation, MPLS Mode is UNIFORM. MSB 3-bits from 8-bits IPv4-ToS or IPv6-TC are copied to the EXP bits of the newly added MPLS header(s).
- During MPLS Decapsulation, MPLS Mode is PIPE. 8-bit IPv4-ToS or IPv6-TC will be retained and hence it provides ToS/TC codepoint transparency.

For the Uniform MPLS mode mapping between IPv4-ToS or IPv6-TC to MPLS-EXP see the table below:

*Mapping between IPv4-TOS/IPv6-TC to MPLS-EXP*

| IPv4-TOS / IPv6-TC | EXP | DSCP |
|---|---|---|
| 0-31 | 0 | 0-7 |
| 32-63 | 1 | 8-15 |
| 64-95 | 2 | 16-23 |
| 96-127 | 3 | 24-31 |
| 128-159 | 4 | 32-39 |
| 160-191 | 5 | 40-47 |
| 192-223 | 6 | 48-55 |
| 224-255 | 7 | 56-63 |

#### Supported Platforms

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

### 6.6.2. HQoS Configuration

#### Basic QoS Configuration Elements

The figure below shows the dependencies between the various QoS configuration elements.

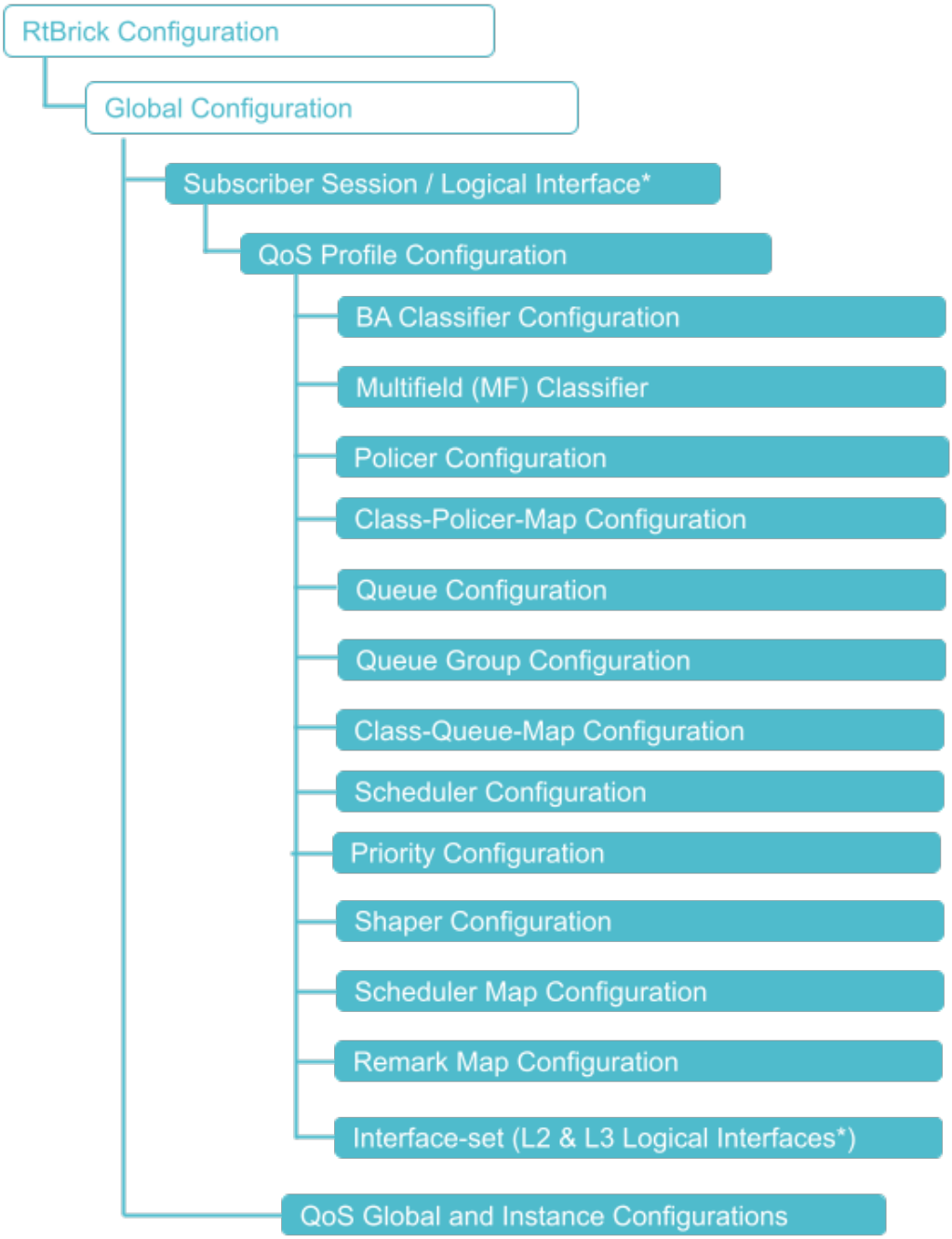



*Figure 140. Dependencies between the various QoS configuration elements*

To configure QoS, perform the following steps which include creating a QoS profile and enabling QoS on a Subscriber-Interface or L3-Interface.

1. Define a QoS profile with classifier, multifield-classifier, class-policer-map, policer, class-queue-map, scheduler-map, and remark-map based on SLAs.
2. Create a Behavioral Aggregate (BA) and/or Multifield (MF) classifier to classify the network traffic at the ingress.
3. Create a policer to police the classified traffic at the ingress with the CIR/PIR defined.
4. Create necessary class-to-policer-map to map the classes to policer levels (mandatory for policing).
5. Create necessary queues with proper size to queue the classified traffic before egressing the

traffic.

6. Create queue groups and define queue numbers (1/4/8).
   NOTE: For the system to initialize with a single queue in a Queue-Group, a system reboot is required.
7. Create the necessary class-to-queue map to map the classes to queues (mandatory for queuing).
8. Specify scheduler(s) with type as required.
9. (Optional) Create the shaper (low-flow / high-flow) and attach it to queue(s) and/or scheduler(s).
10. Specify a scheduler map to define a set of relationships between parent (scheduler or port) and child (queue/queue-group or scheduler) at the egress.
11. (optional) Create Remark-Map for QoS field remarking of the outgoing packet.
12. For the downstream traffic, map the MPLS EXP classifier either to an instance or global entity and/or the Multifield (MF) classifier as a global entity (refer to the figure above.)
13. (optional) Map the MPLS-IPv4/IPv6 remark-map either to an instance or configure it as a global entity.
14. For business services QoS, service-group scheduler is connected to PON scheduler or directly to an IFP.
    NOTE: The service group QoS profile is attached to an interface-set.

> Priority propagation is enabled by default. To disable the Priority Propagation, we recommend doing this at the beginning and not during an active session.

**Hierarchical QoS**

Basic Quality of Service (QoS) is designed to provide a single level of traffic scheduling. In contrast, Hierarchical Quality of Service (HQoS) offers a more sophisticated approach to traffic treatment, utilizing multiple levels of scheduling in a hierarchical manner based on Service Level Agreements (SLAs).

The figure below shows the additional dependencies for Multi-level HQoS.

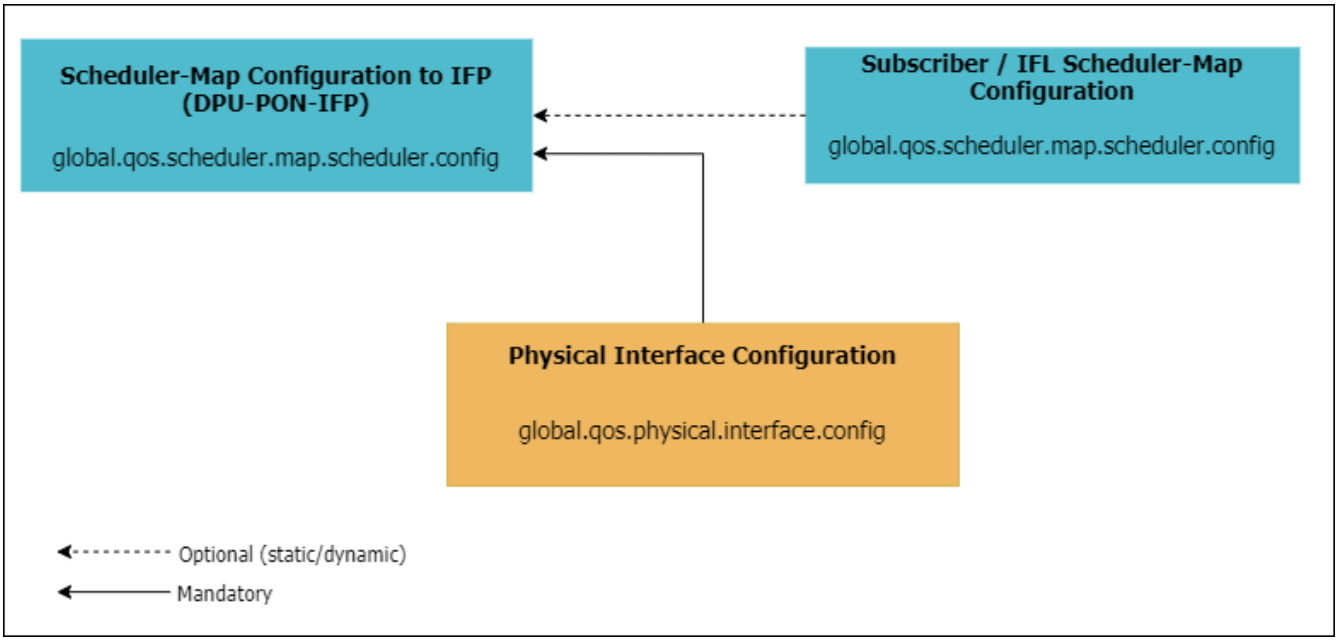



*Figure 141. Dependencies for Multi-level HQoS*

Below are the figures depicting an example of scheduling hierarchy.

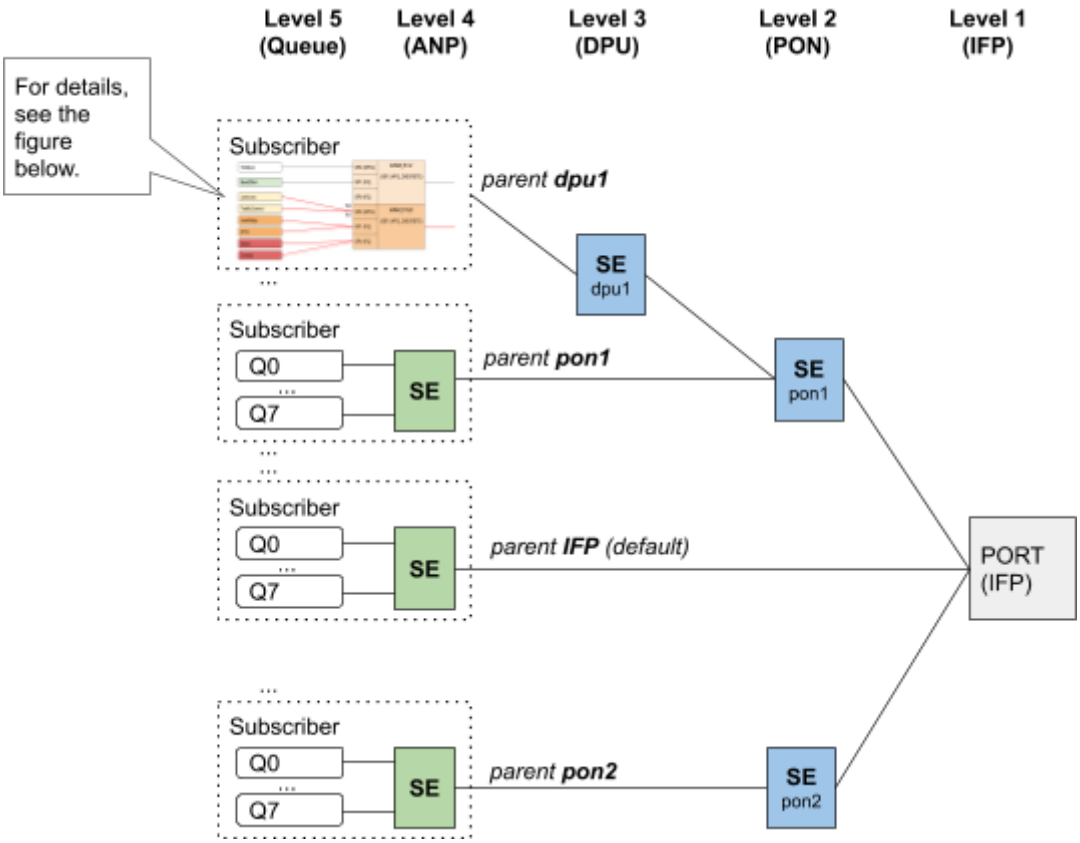



*Figure 142. Example of scheduling hierarchy*

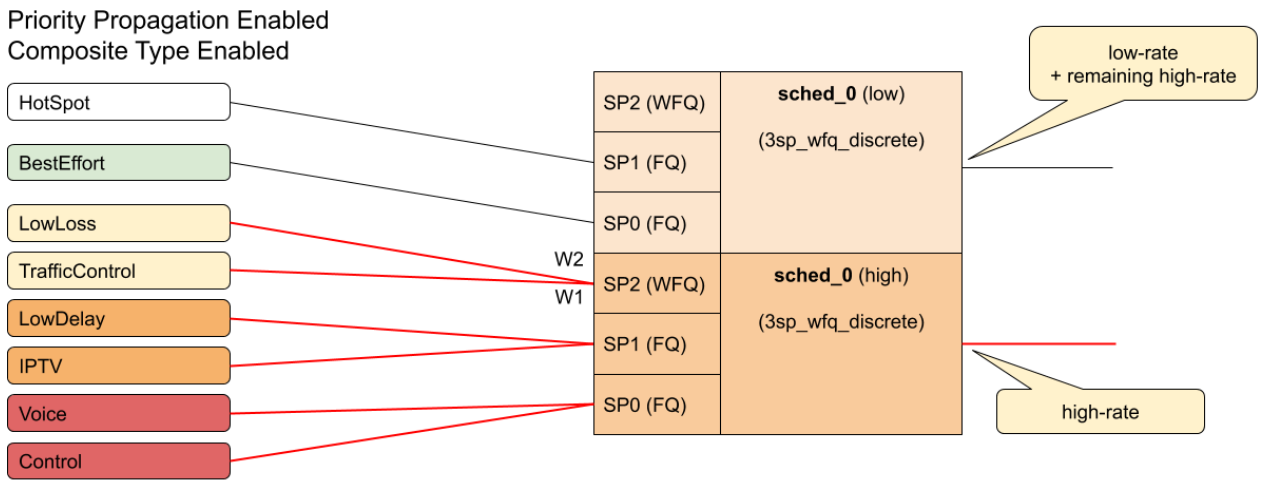



*Figure 143. Example of scheduling hierarchy*

To configure HQoS, map the level-3 to level-5 hierarchy in multi-level HQoS to a different scheduler to represent it and map it to the physical interface.

### Configuration Syntax and Commands

The configurations in this section exemplify the setup of Hierarchical Quality of Service (HQoS) for a subscriber with a residential profile and a Service Level Agreement (SLA) of 20Mbps upstream and downstream. The setup also encompasses four different service types, namely Best Effort(BE), Low-Delay(LD), Low-Loss(LL) and Voice(VO). Different services are assigned to various policer levels for upstream traffic based on their respective traffic classes. Meanwhile, for downstream traffic, these same services are assigned to different queues depending on their traffic class. The traffic class is determined by the configuration of either the BA or MF classifier.

The following sections describe the HQoS configuration syntax and commands.

- Configuring QoS Profiles
- Behavior Aggregate (BA) Classifier Configuration
- Multifield Classifier (MFC) Configuration
- L2 ACL Priority Bit Egress Classifier Configuration
- Policer Configuration
- Class-Policer-Map Configuration
- Queue Configuration
- Queue-Group Configuration
- Class-Queue-Map Configuration
- Scheduler Configuration
- Shaper Configuration
- Scheduler-Map Configuration
- Remark-Map Configuration

## 6.7. Configuring QoS Profiles

A profile configuration defines the QoS profile that is installed on the subscriber or L3 interfaces. The following sections explain the different elements of the QoS profiles that you can create and attach.

Use the following CLI syntax to configure a QoS profile:

**set forwarding-options class-of-service profile** <profile-name> <attribute> <value>

| Attribute | Description |
|---|---|
| `<profile-name>` | Specifies the name of the QoS profile |
| `classifier-name <classifier-name>` | Specifies the name of the BA classifier |
| `multifield-classifier-name <multifield-classifier-name>` | Specifies the name of the multifield classifier |
| `class-policer-map-name <class-policer-map-name>` | Specifies the name of the map that associates a class with a policer level. |
| `policer-name <policer-name>` | Specifies the policer name |
| `class-queue-map-name <class-queue-map-name>` | Specifies the name of the map that associates a class with a queue |
| `scheduler-map-name <scheduler-map-name>` | Specifies the name of the scheduler map |
| `remark-map-name <remark-map-name>` | Specifies the name of the remark map |
| `egress-class-policer-map-name <egress-class-policer-map-name>` | Specifies the name of the egress Class Policer Map.<br>This configuration applies only to the L2X traffic on the LAG interface. |
| `egress-policer-name <egress-policer-name>` | Specifies the name of the egress policer.<br>This configuration applies only to the L2X traffic on the LAG interface. |

In the following example, the QoS profile `residential` is configured with classifier-name `subs-pbit-class`, class-policer-map `policer-map-residential`, policer `policer-residential`, class-queue-map `subs-4queues`, scheduler-map `subs-4queues-residential`, and remark-map `subs-remarking-residential`.

```
set forwarding-options class-of-service profile residential
set forwarding-options class-of-service profile residential classifier-name subs-pbit-class
set forwarding-options class-of-service profile residential class-queue-map-name subs-4queues
set forwarding-options class-of-service profile residential remark-map-name subs-remarking-residential
```

```
set forwarding-options class-of-service profile residential policer-name policer-
residential
set forwarding-options class-of-service profile residential class-policer-map-name
policer-map-residential
set forwarding-options class-of-service profile residential scheduler-map-name subs-
4queues-residential
commit
```

The following example shows the QoS profile configuration of the `residential` profile:

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service
profile
{
  "rtbrick-config:profile": [
    {
      "profile-name": "residential",
      "classifier-name": "subs-pbit-class",
      "class-queue-map-name": "subs-4queues",
      "remark-map-name": "subs-remarking-residential",
      "policer-name": "policer-residential",
      "class-policer-map-name": "policer-map-residential",
      "scheduler-map-name": "subs-4queues-residential"
    }
  ]
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Attaching QoS Profile to IFP

This release supports attaching a QoS profile to a physical interface (IFP).

The following guidelines apply for IFP QoS:

- IFP QoS supports global classifier configurations
- Queue statistics for IFP QoS is supported only in the Q2C-based platforms
- IFP QoS does not support the following configurations:
  - Per-profile classifier configuration
  - Scheduler
  - Remarking
  - Policer
- IFP QoS is not supported on LAG interfaces on Q2C and Q2A platforms

**Attaching QoS Profile to IFP:**

Syntax:

**set interface** <name> **class-of-service** <class-of-service>

| Attribute | Description |
|---|---|
| `<name>` | Name of the physical interface (IFP). |
| `<class-of-service>` | Name of the QoS profile to attach to the IFP. |

The following example assigns the QoS profile named qos_profile_1 to the physical interface ifp-0/1/23, configuring how traffic on that interface is prioritized and shaped.

```
set interface ifp-0/1/23 class-of-service qos_profile_1
```

```
{
  "rtbrick-config:interface": [
      "name: "ifp-0/1/23",
      "class-of-service": "qos_profile_1"
  ]
}
```

#### Behavior Aggregate (BA) Classifier Configuration

Use the following CLI syntax to configure the BA classifier:

**set forwarding-options class-of-service classifier** <classifier-name> <attribute> <value>

| Attribute | Description |
|---|---|
| `<classifier-name>` | Specifies the classifier name |
| `match-type <match-type>` | Specifies the type of traffic to classify, that is, ipv4-tos, ipv6-tc, ieee-802.1, ieee-802.1-inner, exp |
| `match-type <match-type> codepoint <codepoint>` | Specifies the code-point value based on the match-type |
| `match-type <match-type> codepoint <codepoint> class <class>` | Specifies the traffic class as class-0, class-1, class-2, class-3, class-4, class-5, class-6, and class-7 |
| `match-type <match-type> codepoint <codepoint> remark-codepoint <remark-codepoint>` | Specifies the remark-codepoint that is used for remarking |

The following example configures the BA classifier `subs-pbit-class` for traffic `ieee-802.1` with a match code point `2` classified as class `class-1`.

```
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 2 class class-1
```

```
commit
```

The derived class and the expected codepoint must match for the traffic type being classified.

The following example shows BA classifier configuration.

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service
classifier
{
  "rtbrick-config:classifier": [
    {
      "classifier-name": "subs-pbit-class",
      "match-type": [
        {
          "match-type": "ieee-802.1",
          "codepoint": [
            {
              "codepoint": 2,
              "class": "class-1"
            }
          ]
        }
      ]
    }
  ]
}
```

The following configuration sets ieee-802.1-inner classifier match-type.

The inner VLAN P-bit classification feature is available only on the Q2C-based and Q2A-based platforms.

```
set forwarding-options class-of-service classifier subs-inner-pbit
set forwarding-options class-of-service classifier subs-inner-pbit match-type ieee-
802.1-inner
set forwarding-options class-of-service classifier subs-inner-pbit match-type ieee-
802.1-inner codepoint 2
set forwarding-options class-of-service classifier subs-inner-pbit match-type ieee-
802.1-inner codepoint 2 class class-1
```

The following example shows the configuration for the `ieee-802.1-inner` classifier match-type.

```
supervisor@rtbrick>: cfg> show config forwarding-options class-of-service classifier
subs-inner-pbit
{
```

```
  "rtbrick-config:classifier": [
    {
      "classifier-name": "subs-inner-pbit",
      "match-type": [
        {
          "match-type": "ieee-802.1-inner",
          "codepoint": [
            {
              "codepoint": 2,
              "class": "class-1"
            }
          ]
        }
      ]
    }
  ]
}
```

To access the RESTCONF API that corresponds to this CLI, click here.

#### Multifield Classifier (MFC) Configuration

Use the following CLI syntax to configure the multifield classifier.

**set forwarding-options class-of-service multifield-classifier** <attribute> <value>

[NOTE]:

- Starting from Release 20.10.2, Multifield Classifier configuration requires explicit use of ordinal keywords.
- For classification and remarking of best effort upstream traffic, it is recommended to configure an ACL/Classifier that Qualifies the best-effort traffic and updates IP-ToS/IPv6-TC with 0. This ensures that the upstream traffic is marked correctly.
- For best-effort upstream traffic, if no specific ACL is configured in the MFC, it is recommended to configure an ACL with forward class-0 and remark-map 0.

| Attribute | Description |
|---|---|
| `acl <l3v4 \| l3v6> <…>` | Specifies the l3v4 / l3v6 ACL rule for multifield classifier configurations. For more information on configuring the ACL match rules, see the sections below. |

To access the RESTCONF API that corresponds to this CLI, click here.

#### IPv4 ACL Configuration Configuration

Use the following CLI syntax to configure the IPv4 ACL Match Configuration for the multifield

classifier:

**set forwarding-options class-of-service multifield-classifier acl l3v4 rule** <rule-name> ordinal <ordinal-value> <attribute> <Value>

| Attribute | Description |
|---|---|
| `<rule-name>` | Specifies the multifield classifier rule name |
| `ordinal <ordinal-value>` | Specifies the ordinal that is used for traffic policy rule referencing |
| `match <destination-ipv4-prefix>` | Specifies the destination IPv4 prefix address |
| `match <destination-ipv4-prefix-list>` | Specifies the destination IPv4 prefix list name |
| `match <direction>` | Specifies the acl l3 traffic direction match |
| `match <ip-protocol>` | Specifies the IP protocol such as UDP or TCP |
| `match <destination-l4-port>` | Specifies the Layer 4 destination port number |
| `match <ipv4-tos>` | Specifies the IPv4 ToS value |
| `match <ipv4-dscp>` | Specifies the IPv4 Differentiated Services Code Point (DSCP) value |
| `match <forward-class>` | Specifies the forward class name |
| `match <mpls-traffic>` | Specifies the MPLS traffic |
| `match <source-ipv4-prefix>` | Specifies the source IPv4 prefix address |
| `match <source-ipv4-prefix-list>` | Specifies the source IPv4 prefix list name |
| `match <source-l4-port>` | Specifies the Layer 4 source port number |
| `action forward-class <class>` | class-0, class-1, class-2, class-3, class-4, class-5, class-6, class-7 |
| action remark-codepoint <remark-codepoint> | Specifies the remark-map codepoint value. Range: 0-254. |
| action remark-dscp-codepoint <remark-dscp-codepoint> | Specifies the remark DSCP codepoint value. Range: 0-63. To ensure ECN transparency, you must configure the MFC with the `remark-dscp-codepoint` option. Also, the traffic must match and hit the MFC ACL. Remark code-point (`remark-codepoint`) and Remark DSCP code-point (`remark-dscp-codepoint`) are mutually exclusive configurations and cannot be used together. |

Multifield classification commonly matches on IP address fields, the IP protocol type field, or the port number in the UDP or TCP pseudo-header field.

The `l3v4` acl-type multifield classifier is configured with a qualifier/match based on `ipv4-tos` (128, 160) and `source-ipv4-prefix` (132.1.1.3/32) with action as `forward-class` (class-0,class-1).

```
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
```

```
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1001
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1001 match ipv4-tos 128
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1001 match source-ipv4-prefix 132.1.1.3/32
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1001 action forward-class class-0
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1002
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1002 match ipv4-tos 160
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1002 match source-ipv4-prefix 132.1.1.3/32
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1002 action forward-class class-1
commit
```

The following example shows the IPv4 Match Configuration for the multifield classifier:

```
supervisor@rtbrick.net: cfg> show config forwarding-options class-of-service
multifield-classifier acl l3v4 rule global_mfc
{
    "rtbrick-config:rule": [
      {
        "rule-name": "global_mfc",
        "ordinal": [
          {
            "ordinal-value": 1001,
            "match": {
              "ipv4-tos": 128,
              "source-ipv4-prefix": "132.1.1.3/32"
            },
            "action": {
              "forward-class": "class-0"
            }
          },
          {
            "ordinal-value": 1002,
            "match": {
              "ipv4-tos": 160,
              "source-ipv4-prefix": "132.1.1.3/32"
            },
            "action": {
              "forward-class": "class-1"
            }
          }
        ]
      }
    ]
```

```
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### IPv6 ACL Configuration Configuration

Use the following CLI syntax to configure the IPv6 ACL match configuration for the multifield classifier:

**set forwarding-options class-of-service multifield-classifier acl l3v6 rule** <rule-name> ordinal <ordinal-value> <attribute> <value>

| Attribute | Description |
|---|---|
| `<rule-name>` | Specifies the multifield classifier rule name |
| `ordinal <ordinal-value>` | Specifies the ordinal that is used for traffic rule referencing |
| `match <destination-ipv6-prefix>` | Specifies the destination IPv6 prefix address |
| `match <destination-ipv6-prefix-list>` | Specifies the destination IPv6 prefix list name |
| `match <ip-protocol>` | Specifies the IP protocol such as UDP or TCP |
| `match <destination-l4-port>` | Specifies the Layer 4 destination port number |
| `match <ipv6-tc>` | Specifies the IPv6 traffic class value |
| `match <forward-class>` | Specifies the forward class name |
| `match <mpls-traffic>` | Specifies the MPLS traffic |
| `match <source-ipv6-prefix>` | Specifies the source IPv6 prefix address |
| `match <source-ipv6-prefix-list>` | Specifies the source IPv6 prefix list name |
| `match <source-l4-port>` | Specifies the Layer 4 source port number |
| `action forward-class <class>` | class-0, class-1, class-2, class-3, class-4, class-5, class-6, class-7 |
| action remark-codepoint <remark-codepoint> | Specifies the remark-map codepoint value. Range: 0-254. |
| action remark-dscp-codepoint <remark-dscp-codepoint> | Specifies the remark DSCP codepoint value. Range: 0-63. |

In the following example, the `l3v6` acl-type multifield classifier is configured with qualifier/match based on `ipv6-tc` (128, 160) and `source-ipv6-prefix` (132:1:1::3/32) with action as `forward-class` (class-0,class-1).

```
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1001
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1001 match ipv6-tc 128
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1001 match source-ipv6-
prefix 132::1::3/32
```

```
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1001 action forward-class
class-0
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1002
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1002 match ipv6-tc 160
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1002 match source-ipv6-
prefix 132:1:1::3/32
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1002 action forward-class
class-1
commit
```

The following example shows the IPv6 match configuration for the multifield classifier:

```
supervisor@rtbrick.net: cfg> show config forwarding-options class-of-service multifield-classifier acl l3v6 rule
global_mfc
{
    "rtbrick-config:rule": [
      {
        "rule-name": "global_mfc",
        "ordinal": [
          {
            "ordinal-value": 1001,
            "match": {
              "ipv6-tc": 128,
              "source-ipv6-prefix": "132:1:1::3/32"
            },
            "action": {
              "forward-class": "class-0"
            }
          },
          {
            "ordinal-value": 1002,
            "match": {
              "ipv6-tc": 160,
              "source-ipv6-prefix": "132:1:1::3/32"
            },
            "action": {
              "forward-class": "class-1"
            }
          }
        ]
      }
    ]
  }
```

To access the RESTCONF API that corresponds to this CLI, click here.

#### IPv4/IPv6 Priority Configuration

Use the following CLI syntax to configure the IPv4/IPv6 priority configuration:

**set forwarding-options class-of-service multifield-classifier acl [l3v4|l3v6] rule** <rule-name> <attribute> <value>

| Attribute | Description |
|---|---|
| <rule-name> | Specifies the multifield classifier rule name. |

| Attribute | Description |
|---|---|
| ordinal <ordinal-value> | Specifies the ordinal that is used for traffic policy rule referencing. |
| ordinal <ordinal-value> <priority> | Specify the ACL priority value. Range: 100-20000. The default priority value for multifield-classifier ACL is 500. |

The following example configures the `l3v6` acl-type multifield classifier with qualifier/match based on `ipv6-tc` (128) and `source-ipv6-prefix`(132:1:1::3/32) with action as `forward-class` (class-0,class-1) along with `priority` value (100, 200). When there are multiple qualifiers or actions, the one with the higher priority takes precedence over the others.

```
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1001
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1001 match ipv6-tc 128
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1001 match source-ipv6-
prefix 132::1::3/32
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1001 priority 100
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1001 action forward-class
class-0
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1002
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1002 match ipv6-tc 128
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1002 match source-ipv6-
prefix 132:1:1::3/32
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1002 priority 200
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1002 action forward-class
class-1
commit
```

The following example shows the IPv4 priority configuration:

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service multifield-classifier acl l3v4 rule
rtb_mfc


supervisor@rtbrick.net: cfg> show config forwarding-options class-of-service multifield-classifier acl l3v6 rule
global_mfc
{
    "rtbrick-config:rule": [
      {
        "rule-name": "global_mfc",
        "ordinal": [
          {
            "ordinal-value": 1001,
            "priority": 100,
            "match": {
              "ipv6-tc": 128,
              "source-ipv6-prefix": "132:1:1::3/32"
            },
            "action": {
              "forward-class": "class-0"
            }
          },
          {
            "ordinal-value": 1002,
            "priority": 200,
            "match": {
              "ipv6-tc": 160,
              "source-ipv6-prefix": "132:1:1::3/32"
```

```
            },
            "action": {
              "forward-class": "class-1"
            }
          }
        ]
      }
    ]
  }
```

To access the RESTCONF API that corresponds to this CLI, click here.

### 6.7.1. L2 ACL Priority Bit Egress Classifier Configuration

Use the following CLI syntax to configure the L2 ACL priority bit egress classifier.

**set forwarding-options** <attribute> <value>

| Attribute | Description |
|---|---|
| `acl l2 <…>` | Specifies the l2 ACL rule for egress priority bit classifier configurations. For more information on configuring the ACL match rules, see the sections below. |

#### Service VLAN Priority Configuration

Use the following CLI syntax to configure the L2 ACL Match Configuration for the egress priority bit classifier:

**set forwarding-options acl l2 rule** <rule-name> ordinal <ordinal-value> <attribute> <Value>

| Attribute | Description |
|---|---|
| `<rule-name>` | Specifies the L2 ACL classifier rule name |
| `ordinal <ordinal-value>` | Specifies the ordinal that is used for traffic rule referencing |
| `match <direction>` | Specifies the direction of L2 traffic |
| `match <service-outer-vlan-priority>` | Specifies the outer VLAN priority of the L2 service traffic |
| `action forward-class <class>` | class-0, class-1, class-2, class-3, class-4, class-5, class-6, class-7 |

L2 ACL classification commonly matches on layer-2 fields such as service-inner-vlan-priority, service-outer-vlan-priority, interface-meta-data, and direction

The l2 acl type egress priority bit classifier is configured with a qualifier/match based on service-inner-vlan-priority ` (0-7), interface-meta-data (type-1), direction (ingress), `service-outer-vlan-priority` (0-7) with action as `forward-class` (class-0 to class-7).

```
set forwarding-options acl l2 rule evpn_vpws ordinal 0
```

```
set forwarding-options acl l2 rule evpn_vpws ordinal 0 match direction ingress
set forwarding-options acl l2 rule evpn_vpws ordinal 0 match service-outer-vlan-priority 0
set forwarding-options acl l2 rule evpn_vpws ordinal 0 match interface-meta-data type-1
set forwarding-options acl l2 rule evpn_vpws ordinal 0 action forward-class class-0
set forwarding-options acl l2 rule evpn_vpws ordinal 1
set forwarding-options acl l2 rule evpn_vpws ordinal 1 match direction ingress
set forwarding-options acl l2 rule evpn_vpws ordinal 1 match service-outer-vlan-priority 1
set forwarding-options acl l2 rule evpn_vpws ordinal 1 match interface-meta-data type-1
set forwarding-options acl l2 rule evpn_vpws ordinal 1 action forward-class class-1
commit
```

The following example shows the L2 ACL P-Bit Egress Classifier configuration:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config forwarding-options acl l2 rule evpn_vpws ordinal
{
  "rtbrick-config:ordinal": [
    {
      "ordinal-value": 0,
      "match": {
        "direction": "ingress",
        "service-outer-vlan-priority": 0,
        "interface-meta-data": "type-1"
      },
      "action": {
        "forward-class": "class-0"
      }
    },
    {
      "ordinal-value": 1,
      "match": {
        "direction": "ingress",
        "service-outer-vlan-priority": 1,
        "interface-meta-data": "type-1"
      },
      "action": {
        "forward-class": "class-1"
      }
    }
  ]
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Policer Configuration

Use the following CLI syntax to configure the QoS policer:

**set forwarding-options class-of-service policer** <policer-name> <attribue> <value>

| Attribute | Description |
|---|---|
| `<policer-name>` | Specifies the policer name. |

| Attribute | Description |
|---|---|
| levels <levels> | Specifies levels in the policer. There is only support for the policer levels 1 and 4.<br><br>If the QoS policer levels is set to 1, the rate configuration for levels 2, 3, and 4 is blocked. Changing levels from 4 to 1 also blocks the configuration, unless only one level rate is set. |
| `type <type>` | Specifies the policer type. |
| `flags <flag>` | Specifies the policer flags. |
| `level1-rates cir <cir>` | Specifies the committed information rate (CIR) in Kilobits per second (Kbps) for level-1. The same is applicable for level-2 to level-4. |
| `level1-rates pir <pir>` | Specifies the peak information rate (PIR) in Kilobits per second (Kbps) for level-1. The same is applicable for level-2 to level-4. |
| `level1-rates cbs <cbs>` | Specifies the Committed burst size (CBS) in Kilobits per second (Kbps) for level-1. The same is applicable for level-2 to level-4. |
| `level1-rates pbs <pbs>` | Specifies the peak burst size (PBS) in Kilobits per second (Kbps) for level-1. The same is applicable for level-2 to level-4. |
| `level1-rates max-cir <max-cir>` | Specifies the maximum for the level-1 committed information rate (CIR) in kilobits per second (Kbps). The same is applicable for level-2 to level-4. |
| `level1-rates max-pir <max-pir>` | Specifies the maximum for the level-1 peak information rate (PIR) in Kilobits per second (Kbps). The same is applicable for level-2 to level-4. |
| `mode <single-level \| large-bucket \| single-level-large-bucket>` | Specifies the mode of the QoS policer. Supported modes: `large-bucket`, `single-level`, and `single-level-large-bucket`. The Committed burst size (CBS) and Peak burst size(PBS) can be specified in the range 0-2088960 (kbits). The burst size of the policer is configurable up to 4 MB in regular mode. The `large-bucket` mode allows the configuration of buckets ranging from 4 MB to 255 MB.<br><br>• Operating in large-bucket mode may reduce the policer rate accuracy. |

The following example configures the QoS policer `policer-residential` with multi-level policers (levels=4), CIR, PIR rates, and burst sizes (CBS, PBS) for each level (level-1, level-2, level-3 and level-4). Also, `two-rate-three-color` policer type and color-blind as the default configuration. The four-level Policer configuration is as follows:

- Level-1(cir=2Mbps, pir=2.5Mbps, cbs=1000, pbs=1000)
- Level-2(cir=3Mbps, pir=3.5Mbps, cbs=1000, pbs=1000)

- Level-3(cir=4Mbps, pir=4.5Mbps, cbs=1000, pbs=1000)
- Level-4(cir=1Mbps, pir=1.5Mbps, cbs=1000, pbs=1000)

The following configurations show policer configurations for levels=4.

```
set forwarding-options class-of-service policer policer-residential
set forwarding-options class-of-service policer policer-residential level1-rates cir 2000
set forwarding-options class-of-service policer policer-residential level1-rates cbs 1000
set forwarding-options class-of-service policer policer-residential level1-rates pir 2500
set forwarding-options class-of-service policer policer-residential level1-rates pbs 1000
set forwarding-options class-of-service policer policer-residential level2-rates cir 3000
set forwarding-options class-of-service policer policer-residential level2-rates cbs 1000
set forwarding-options class-of-service policer policer-residential level2-rates pir 3500
set forwarding-options class-of-service policer policer-residential level2-rates pbs 1000
set forwarding-options class-of-service policer policer-residential level3-rates cir 4000
set forwarding-options class-of-service policer policer-residential level3-rates cbs 1000
set forwarding-options class-of-service policer policer-residential level3-rates pir 4500
set forwarding-options class-of-service policer policer-residential level3-rates pbs 1000
set forwarding-options class-of-service policer policer-residential level4-rates cir 1000
set forwarding-options class-of-service policer policer-residential level4-rates cbs 1000
set forwarding-options class-of-service policer policer-residential level4-rates pir 1500
set forwarding-options class-of-service policer policer-residential level4-rates pbs 1000
set forwarding-options class-of-service policer policer-residential levels 4
set forwarding-options class-of-service policer policer-residential type two-rate-three-color
commit
```

The following example shows the QoS policer level (levels=4) rates configuration:

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service policer policer-residential
{
  "rtbrick-config:policer": [
{
            "policer-name": "policer-residential",
            "level1-rates": {
              "cir": 2000,
              "cbs": 1000,
              "pir": 2500,
              "pbs": 1000
            },
            "level2-rates": {
              "cir": 3000,
              "cbs": 1000,
              "pir": 3500,
              "pbs": 1000
            },
            "level3-rates": {
              "cir": 4000,
              "cbs": 1000,
              "pir": 4500,
              "pbs": 1000
            },
            "level4-rates": {
              "cir": 1000,
              "cbs": 1000,
```

```
              "pir": 1500,
              "pbs": 1000
            },
            "levels": 4,
            "type": "two-rate-three-color"
          }
  ]
}
```

The following configurations show policer configurations for levels=1.

```
set forwarding-options class-of-service policer policer-residential
set forwarding-options class-of-service policer policer-residential level1-rates cir 2000
set forwarding-options class-of-service policer policer-residential level1-rates cbs 1000
set forwarding-options class-of-service policer policer-residential level1-rates pir 2500
set forwarding-options class-of-service policer policer-residential level1-rates pbs 1000
set forwarding-options class-of-service policer policer-residential levels 1
set forwarding-options class-of-service policer policer-residential type two-rate-three-color
commit
```

The following example shows the QoS policer level rates (levels=1) configuration:

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service policer policer-residential
{
  "rtbrick-config:policer": [
{
            "policer-name": "policer-residential",
            "level1-rates": {
              "cir": 2000,
              "cbs": 1000,
              "pir": 2500,
              "pbs": 1000
            }
            "levels": 1,
            "type": "two-rate-three-color"
          }
  ]
}
```

The following example shows the Large Bucket mode configuration in QoS Policer.

```
set forwarding-options class-of-service policer policer-residential
set forwarding-options class-of-service policer policer-residential level1-rates
set forwarding-options class-of-service policer policer-residential level1-rates cir 700000000
set forwarding-options class-of-service policer policer-residential level1-rates cbs 2088960
set forwarding-options class-of-service policer policer-residential level1-rates pir 700000000
set forwarding-options class-of-service policer policer-residential level1-rates pbs 2088960
set forwarding-options class-of-service policer policer-residential level2-rates
set forwarding-options class-of-service policer policer-residential level2-rates cir 3000
set forwarding-options class-of-service policer policer-residential level2-rates cbs 1000
set forwarding-options class-of-service policer policer-residential level2-rates pir 3500
set forwarding-options class-of-service policer policer-residential level2-rates pbs 1000
```

```
set forwarding-options class-of-service policer policer-residential level3-rates
set forwarding-options class-of-service policer policer-residential level3-rates cir 4000
set forwarding-options class-of-service policer policer-residential level3-rates cbs 1000
set forwarding-options class-of-service policer policer-residential level3-rates pir 4500
set forwarding-options class-of-service policer policer-residential level3-rates pbs 1000
set forwarding-options class-of-service policer policer-residential level4-rates
set forwarding-options class-of-service policer policer-residential level4-rates cir 1000
set forwarding-options class-of-service policer policer-residential level4-rates cbs 1000
set forwarding-options class-of-service policer policer-residential level4-rates pir 1500
set forwarding-options class-of-service policer policer-residential level4-rates pbs 1000
set forwarding-options class-of-service policer policer-residential levels 4
set forwarding-options class-of-service policer policer-residential type two-rate-three-color
set forwarding-options class-of-service policer policer-residential mode large-bucket
```

API To access the RESTCONF API that corresponds to this CLI, click here.

### 6.7.2. Class-Policer-Map Configuration

Use the following CLI syntax to configure the Class-Policer-Map:

**set forwarding-options class-of-service class-policer-map** <class-policer-map-name> <attribue> <value>

| Attribute | Description |
|---|---|
| `<class-policer-map-name>` | Specifies the class policer map name, |
| `class <class>` | Specifies the class such as class-0, class-1, class-2, class-3, class-4, class-5, class-6, class-7 |
| `class <class> <policer-level>` | Specifies the policer levels. The supported levels are: level-1, level-2, level-3, and level-4 |

Below is an example configuration of the class-policer-map, which sets up level-1 to level-4 policer levels.

- class-0 mapped to policer level-1
- class-1 mapped to policer level-2
- class-2 mapped to policer level-3
- class-3 mapped to policer level-4

```
set forwarding-options class-of-service class-policer-map policer-map-residential class class-0 policer-level level-1
set forwarding-options class-of-service class-policer-map policer-map-residential class class-1 policer-level level-2
set forwarding-options class-of-service class-policer-map policer-map-residential class class-2 policer-level level-3
set forwarding-options class-of-service class-policer-map policer-map-residential class class-3 policer-level level-4
commit
```

The following example shows the class-policer-map configuration:

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service class-policer-map policer-map-l2tp
{
  "rtbrick-config:class-policer-map": [
{
            "class-policer-map-name": "policer-map-residential",
            "class": [
              {
                "class": "class-0",
                "policer-level": "level-1"
              },
              {
                "class": "class-1",
                "policer-level": "level-2"
              },
              {
                "class": "class-2",
                "policer-level": "level-3"
              },
              {
                "class": "class-3",
                "policer-level": "level-4"
              }
            ]
          }
  ]
}
```

To access the RESTCONF API that corresponds to this CLI, click here.

#### Queue Configuration

Use the following CLI syntax to configure a queue:

**set forwarding-options class-of-service queue** <queue-name> <atribute> <value>

| Attribute | Description |
|---|---|
| `<queue-name>` | Specifies the queue name |
| `queue-size <queue-size>` | Specifies the size of the queue in bytes |
| `<shaper-name>` | (Optional) Specifies the shaper that is associated with the queue. |
| `wred minimum-threshold <minimum-threshold>` | Specifies the minimum average queue size to apply WRED in bytes |
| `wred maximum-threshold <maximum-threshold>` | Specifies the maximum average queue size to apply WRED in bytes |
| `wred drop-probability <drop-probability>` | WRED drop probability applied at the maximum threshold |
| `header-compensation bytes <bytes>` | Specifies the header compensation value |
| `header-compensation decrement true` | Specifies whether the header compensation value is to be decremented. |

The following example configures four queues with different traffic streams named BE_SUBS, LD_SUBS, LL_SUBS and VO_SUBS with queue attributes queue-size, queue header-compensation & queue shaper for Best Effort(BE), Low-Delay(LD), Low-Loss(LL) and Voice(VO) Queues.

```
set forwarding-options class-of-service queue BE_SUBS
set forwarding-options class-of-service queue BE_SUBS queue-size 375000
set forwarding-options class-of-service queue BE_SUBS header-compensation bytes 22
set forwarding-options class-of-service queue BE_SUBS header-compensation decrement true
set forwarding-options class-of-service queue LD_SUBS
set forwarding-options class-of-service queue LD_SUBS queue-size 625000
set forwarding-options class-of-service queue LD_SUBS header-compensation bytes 22
set forwarding-options class-of-service queue LD_SUBS header-compensation decrement true
set forwarding-options class-of-service queue LL_SUBS
set forwarding-options class-of-service queue LL_SUBS queue-size 625000
set forwarding-options class-of-service queue LL_SUBS header-compensation bytes 22
set forwarding-options class-of-service queue LL_SUBS header-compensation decrement true
set forwarding-options class-of-service queue VO_SUBS
set forwarding-options class-of-service queue VO_SUBS queue-size 156250
set forwarding-options class-of-service queue VO_SUBS header-compensation bytes 22
set forwarding-options class-of-service queue VO_SUBS header-compensation decrement true
set forwarding-options class-of-service queue VO_SUBS shaper-name shaper_VO
commit
```

The following example shows the queue configuration:

```
supervisor@DT-STD-23-2402>bm14-tst.fsn.rtbrick.net: cfg> show config forwarding-options class-of-service queue
{
  "rtbrick-config:queue": [
{
            "queue-name": "BE_SUBS",
            "queue-size": 375000,
            "header-compensation": {
              "bytes": 22,
              "decrement": "true"
            }
          },
          {
            "queue-name": "LD_SUBS",
            "queue-size": 625000,
            "header-compensation": {
              "bytes": 22,
              "decrement": "true"
            }
          },
          {
            "queue-name": "LL_SUBS",
            "queue-size": 625000,
            "header-compensation": {
              "bytes": 22,
              "decrement": "true"
            }
          },
          {
            "queue-name": "VO_SUBS",
```

```
            "queue-size": 156250,
            "shaper-name": "shaper_VO",
            "header-compensation": {
              "bytes": 22,
              "decrement": "true"
            }
          }
  ]
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Queue-Group Configuration

### Queue group size: 1 or 4 or 8

Use the following CLI syntax to configure a queue group:

**set forwarding-options class-of-service queue-group** <queue-group-name> <attribute> <value>

| Attribute | Description |
|---|---|
| `<queue-group-name>` | User-defined name for the queue-group |
| `queue-numbers <queue-numbers>` | Specifies the number of queues in a Queue Group. Supported queue group size is: 1 or 4 or 8. |

The following examples configure the queue group with queue numbers 1 and 4.

```
set forwarding-options class-of-service queue-group subs-4queues queue-numbers 1
commit
```

The following example shows the queue group configuration:

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service queue-group
{
  "rtbrick-config:queue-group": [
    {
      "queue-group-name": "subs-4queues",
      "queue-numbers": 1
    }
  ]
}
```

```
set forwarding-options class-of-service queue-group subs-4queues queue-numbers 4
```

The following example shows the queue group configuration:

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service queue-group
{
  "rtbrick-config:queue-group": [
    {
      "queue-group-name": "subs-4queues",
      "queue-numbers": 4
    }
  ]
}
```

To access the RESTCONF API that corresponds to this CLI, click here.

#### Class-Queue-Map Configuration

Use the following CLI syntax to configure the class-queue-map:

**set forwarding-options class-of-service class-queue-map** <class-queue-map-name> <attribute> <value>

| Attribute | Description |
|---|---|
| `<class-queue-map-name>` | Specifies the class queue map name |
| `class <class>` | Specifies the class such as class-0, class-1, class-2, class-3, class-4, class-5, class-6, class-7 |
| `class <class> queue-name <queue-name>` | Specifies the queue name |

The following example configures the class-queue-map for the specific classes and queues:

- class-0 mapped to queue BE_SUBS
- class-1 mapped to queue LD_SUBS
- class-2 mapped to queue LL_SUBS
- class-3 mapped to queue VO_SUBS

```
set forwarding-options class-of-service class-queue-map subs-4queues class class-0 queue-name BE_SUBS
set forwarding-options class-of-service class-queue-map subs-4queues class class-1 queue-name LD_SUBS
set forwarding-options class-of-service class-queue-map subs-4queues class class-2 queue-name LL_SUBS
set forwarding-options class-of-service class-queue-map subs-4queues class class-3 queue-name VO_SUBS
commit
```

The following example shows the class-queue-map configuration:

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service class-queue-map
{
  "rtbrick-config:class-queue-map": [
{
```

```
            "class-queue-map-name": "subs-4queues",
            "class": [
              {
                "class-type": "class-0",
                "queue-name": "BE_SUBS"
              },
              {
                "class-type": "class-1",
                "queue-name": "LD_SUBS"
              },
              {
                "class-type": "class-2",
                "queue-name": "LL_SUBS"
              },
              {
                "class-type": "class-3",
                "queue-name": "VO_SUBS"
              }
            ]
          }
  ]
}
```

To access the RESTCONF API that corresponds to this CLI, click here.

#### Scheduler Configuration

Use the following CLI syntax to configure a scheduler:

**set forwarding-options class-of-service scheduler** <attribue> <value>

| Attribute | Description |
|---|---|
| `<scheduler-name>` | User-defined Scheduler Name |
| `<scheduler-name> shaper-name <shaper-name>` | (Optional) User-defined Shaper Name |
| `<scheduler-name> type <type>` | Specifies the Scheduler Type<br>2sp_wfq_discrete<br>3sp_wfq_discrete<br>strict_priority<br>wfq_discrete_2sp<br>2sp_wfq_independent<br>fair_queueing<br>weighted_fair_queueing<br>wfq_independent_2sp |
| `<scheduler-name> composite true` | (Optional) keyword to specify the scheduler as composite type |

The following example configures two schedulers: `subs-4queues` with scheduler type

`strict_priority` for the Subscriber level, and `pon0` with scheduler type `fair_queueing` for PON/GPON level.

```
set forwarding-options class-of-service scheduler subs-4queues
set forwarding-options class-of-service scheduler subs-4queues shaper shaper_session
set forwarding-options class-of-service scheduler subs-4queues type strict_priority
set forwarding-options class-of-service scheduler pon0
set forwarding-options class-of-service scheduler pon0 type fair_queueing
commit
```

The following example shows the QoS scheduler configuration:

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service scheduler
{
  "rtbrick-config:scheduler": [
          {
            "scheduler-name": "pon0",
            "type": "fair_queueing"
          },
          {
            "scheduler-name": "subs-4queues",
            "shaper-name": "shaper_session",
            "type": "strict_priority"
          }
  ]
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Priority Propagation

Use the following CLI syntax to configure priority propagation:

**set forwarding-options class-of-service global priority-propagation [enable | disable]**

The following example enables priority propagation at the global level to operate schedulers in dual-flow mode, with high-priority and low-priority flows.

```
set forwarding-options class-of-service global priority-propagation enable
```

The following example shows the priority propagation:

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service global
{
  "rtbrick-config:global": {
    "priority-propagation": "enable"
```

```
  }
}
```

#### Shaper Configuration

Use the following CLI syntax to configure a shaper:

**set forwarding-options class-of-service shaper** <shaper-name> <attribue> <value>

### Command Arguments

| Attribute | Description |
|---|---|
| `<shaper-name>` | User-defined shaper name |
| `<shaper-name> shaping-rate-high <0-700000000>` | High flow shaping rate in kilobits per second |
| `<shaper-name> shaping-rate-high-precision true` | Configures the precision of high-flow shapers, which can only be used with session shapers. |
| `<shaper-name> shaping-rate-low <0-700000000>` | Low flow shaping rate in kilobits per second |

- Low-rate shaping is only supported for high-priority flows. Enabling low-rate shaping consumes additional QoS resources, which will impact the scalability.
- If priority propagation is not enabled, high-flow shaping value will be considered for shaper.
- If the scheduler type is `strict_priority`, the mapping of queues to priorities begins with `strict_priority_1`.
- Shaper rate may not be accurate for fragmented data traffic.

The following example configures the shaper high-flow and low-flow rates for Subscriber Session Level with shaper-name `shaper_session` and Queue Level with shaper-name `shaper_VO`.

```
set forwarding-options class-of-service shaper shaper_session
set forwarding-options class-of-service shaper shaper_session shaping-rate-high 10000
set forwarding-options class-of-service shaper shaper_session shaping-rate-low 100
set forwarding-options class-of-service shaper shaper_VO
set forwarding-options class-of-service shaper shaper_VO shaping-rate-high 2000
set forwarding-options class-of-service shaper shaper_VO shaping-rate-low 0
commit
```

The following example shows the shaper configuration:

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service shaper
```

```
{
  "rtbrick-config:shaper": [
          {
            "shaper-name": "shaper_VO",
            "shaping-rate-high": 2000,
            "shaping-rate-low": 0
          },
          {
            "shaper-name": "shaper_session",
            "shaping-rate-high": 10000,
            "shaping-rate-low": 100
          }
  ]
}
```

With the following shaper configuration, the low flow is limited to 1Mbps and the high flow is limited to 512Kbps. This results in a session rate of approximately 1.5Mbps when 10Mbps is distributed between high and low flows.

```
set forwarding-options class-of-service shaper session-shaper
set forwarding-options class-of-service shaper session-shaper shaping-rate-high-
precision true
set forwarding-options class-of-service shaper session-shaper shaping-rate-high
512
set forwarding-options class-of-service shaper session-shaper shaping-rate-low
1000
```

The following example shows the shaper configuration, where the high precision shaping is enabled:

```
{
  "rtbrick-config:shaper": [
    {
      "shaper-name": "session-shaper",
      "shaping-rate-high": 512,
      "shaping-rate-high-precision": "true",
      "shaping-rate-low": 1000
   }
  ]
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Scheduler-Map Configuration

Use the following CLI syntax to configure a Scheduler-Map:

**set forwarding-options class-of-service scheduler-map** <scheduler-map-name> <attribue> <value>

### Command arguments

| Attribute | Description |
|---|---|
| `<scheduler-map-name>` | Specifies the name of the scheduler-map |
| `scheduler-name <scheduler-name>` | Specifies the name of the scheduler |
| `queue-group-name <group-name>` | Specifies the name of the queue-group |
| `queue-group-name <group-name> queue-name <name>` | Specifies the name of the queue |
| `queue-group-name <group-name> queue-name <name> connection-point <connection-point>` | Specifies the type of connection point, such as no_priority, strict_priority_0, strict_priority_1, strict_priority_2, strict_priority |
| `queue-group-name <group-name> queue-name <name> parent-flow <high-flow \/ low-flow>` | (Optional) Specifies the type of the parent flow, that is high-flow or low-flow. |
| `queue-group-name <group-name> queue-name <name> parent-scheduler-name <parent-scheduler-name>` | Specifies the name of the parent scheduler |
| `queue-group-name <group-name> queue-name <name> port-connection <port-connection-type>` | Specifies the type of port connection, that is, queue_to_port or scheduler_to_port |
| `queue-group-name <group-name> queue-name <name> weight <weight>` | Specifies a scheduling weight for the specified queue. |

The following example configures a scheduler map for OLT with `schedmap-olt` and for subscribers with `subs-4queues-residential`. OLT scheduler-map `schedmap-olt` is directly connected to the physical port and `subs-4queues-residential` connects different queues (BE_SUBS, LL_SUBS,LD_SUBS,VO_SUBS) with different connection-points(strict_priority_3, strict_priority_2, strict_priority_1, strict_priority_0).

```
set forwarding-options class-of-service scheduler-map schedmap-olt
set forwarding-options class-of-service scheduler-map schedmap-olt scheduler-name pon0
set forwarding-options class-of-service scheduler-map schedmap-olt scheduler-name pon0 port-connection scheduler_to_port
```

```
set forwarding-options class-of-service scheduler-map subs-4queues-residential
set forwarding-options class-of-service scheduler-map subs-4queues-residential queue-group-name subs-4queues
set forwarding-options class-of-service scheduler-map subs-4queues-residential queue-group-name subs-4queues queue-name
BE_SUBS
set forwarding-options class-of-service scheduler-map subs-4queues-residential queue-group-name subs-4queues queue-name
BE_SUBS parent-flow high-flow
set forwarding-options class-of-service scheduler-map subs-4queues-residential queue-group-name subs-4queues queue-name
BE_SUBS parent-scheduler-name subs-4queues
set forwarding-options class-of-service scheduler-map subs-4queues-residential queue-group-name subs-4queues queue-name
BE_SUBS connection-point strict_priority_3
set forwarding-options class-of-service scheduler-map subs-4queues-residential queue-group-name subs-4queues queue-name
LD_SUBS
set forwarding-options class-of-service scheduler-map subs-4queues-residential queue-group-name subs-4queues queue-name
LD_SUBS parent-flow high-flow
set forwarding-options class-of-service scheduler-map subs-4queues-residential queue-group-name subs-4queues queue-name
LD_SUBS parent-scheduler-name subs-4queues
set forwarding-options class-of-service scheduler-map subs-4queues-residential queue-group-name subs-4queues queue-name
LD_SUBS connection-point strict_priority_1
set forwarding-options class-of-service scheduler-map subs-4queues-residential queue-group-name subs-4queues queue-name
LL_SUBS
set forwarding-options class-of-service scheduler-map subs-4queues-residential queue-group-name subs-4queues queue-name
LL_SUBS parent-flow high-flow
set forwarding-options class-of-service scheduler-map subs-4queues-residential queue-group-name subs-4queues queue-name
LL_SUBS parent-scheduler-name subs-4queues
set forwarding-options class-of-service scheduler-map subs-4queues-residential queue-group-name subs-4queues queue-name
LL_SUBS connection-point strict_priority_2
set forwarding-options class-of-service scheduler-map subs-4queues-residential queue-group-name subs-4queues queue-name
VO_SUBS
set forwarding-options class-of-service scheduler-map subs-4queues-residential queue-group-name subs-4queues queue-name
VO_SUBS parent-flow high-flow
set forwarding-options class-of-service scheduler-map subs-4queues-residential queue-group-name subs-4queues queue-name
VO_SUBS parent-scheduler-name subs-4queues
set forwarding-options class-of-service scheduler-map subs-4queues-residential queue-group-name subs-4queues queue-name
VO_SUBS connection-point strict_priority_0
set forwarding-options class-of-service scheduler-map subs-4queues-residential scheduler-name subs-4queues
set forwarding-options class-of-service scheduler-map subs-4queues-residential scheduler-name subs-4queues port-
connection scheduler_to_port
commit
```

The following example shows the scheduler-map configuration:

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service scheduler-map
{
  "rtbrick-config:scheduler-map": [
{
            "scheduler-map-name": "schedmap-olt",
            "scheduler-name": [
              {
                "name": "pon0",
                "port-connection": "scheduler_to_port"
              }
            ]
          },
          {
            "scheduler-map-name": "subs-4queues-residential",
            "queue-group-name": [
              {
                "group-name": "subs-4queues",
                "queue-name": [
                  {
```

```
                "name": "BE_SUBS",
                "parent-flow": "high-flow",
                "parent-scheduler-name": "subs-4queues",
                "connection-point": "strict_priority_3"
              },
              {
                "name": "LD_SUBS",
                "parent-flow": "high-flow",
                "parent-scheduler-name": "subs-4queues",
                "connection-point": "strict_priority_1"
              },
              {
                "name": "LL_SUBS",
                "parent-flow": "high-flow",
                "parent-scheduler-name": "subs-4queues",
                "connection-point": "strict_priority_2"
              },
              {
                "name": "VO_SUBS",
                "parent-flow": "high-flow",
                "parent-scheduler-name": "subs-4queues",
                "connection-point": "strict_priority_0"
              }
            ]
          }
        ],
        "scheduler-name": [
          {
            "name": "subs-4queues",
            "port-connection": "scheduler_to_port"
          }
        ]
      }
  ]
}
```


API To access the RESTCONF API that corresponds to this CLI, click here.

#### Remark-Map Configuration

Use the following CLI syntax to configure the remark-map:

**set forwarding-options class-of-service remark-map** <remark-map-name> <attribute> <value>

### Command arguments

| Attribute | Description |
|---|---|
| `<remark-map-name>` | Specifies the remaking map name. |

| Attribute | Description |
|---|---|
| `remark-type <remark-type>` | Specifies the remarking type - ipv4-tos, ipv6-tc, mpls-ipv4, mpls-ipv6, ieee-802.1, class-to-ip, ip-tos.<br><br>The "ip-tos" remark-type can be used to set the IPv4 ToS or IPv6 TC values in IPv4 or IPv6 packet header respectively. |
| `remark-type <remark-type> <match-codepoint>` | Specifies the match code point for the specified remarking type.<br><br>On the UfiSpace S9600-72XC and UfiSpace S9600-32X platforms, the match codepoint is TOS for VLAN IEEE-802.1p remarking. |
| `remark-type <remark-type> <match-codepoint> color <color>` | Indicates the color - all, green, yellow. Color is used to set different remark codepoints for the same match-codepoint based on color marked by the Policer. |
| `remark-type <remark-type> <match-codepoint> color <color> remark-codepoint <remark-codepoint>` | Specifies the remarking codepoint |

To manage best-effort upstream traffic when no specific Access Control List is configured in the MFC, it is recommended to apply an ACL that explicitly assigns forwarding-class 0 and remark-map 0. This ensures consistent traffic prioritization for all default flows.

In the following example, the remark map `subs-remarking-residential` is configured with match-codepoint as `128`, `160`, `192`, and `224`. The color is set to "all", and the remark-codepoint is `6`.

```
set forwarding-options class-of-service remark-map subs-remarking-residential
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-codepoint 128
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-codepoint 128
color all
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-codepoint 128
color all remark-codepoint 6
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-codepoint 160
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-codepoint 160
color all
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-codepoint 160
color all remark-codepoint 6
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-codepoint 192
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-codepoint 192
color all
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-codepoint 192
color all remark-codepoint 6
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-codepoint 224
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-codepoint 224
color all
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-codepoint 224
color all remark-codepoint 6
commit
```

The following example shows the remark map configuration:

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service remark-map remark-exp remark-type ipv6-tc
{
  "rtbrick-config:remark-type": [
{
            "remark-map-name": "subs-remarking-residential",
            "remark-type": [
              {
                "remark-type": "ieee-802.1",
                "match-codepoint": [
                  {
                    "match-codepoint": 128,
                    "color": [
                      {
                        "color": "all",
                        "remark-codepoint": 6
                      }
                    ]
                  },
                  {
                    "match-codepoint": 160,
                    "color": [
                      {
                        "color": "all",
                        "remark-codepoint": 6
                      }
                    ]
                  },
                  {
                    "match-codepoint": 192,
                    "color": [
                      {
                        "color": "all",
                        "remark-codepoint": 6
                      }
                    ]
                  },
                  {
                    "match-codepoint": 224,
                    "color": [
                      {
                        "color": "all",
                        "remark-codepoint": 6
                      }
                    ]
                  }
                ]
              }
            ]
          }
  ]
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Global Profile Mapping

Use the following CLI syntax to configure the remark map for a global profile.

**set forwarding-options class-of-service global remark-map-name** <remark-map-name>

### Command arguments

| Attribute | Description |
|---|---|
| `<remark-map-name>` | Specifies the name of the remark map. |

The following example configures the remark map for a global profile:

```
set forwarding-options class-of-service global remark-map-name subs-remarking-residential
```

The following example shows the remark map for a global profile configuration:

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service global
{
  "rtbrick-config:global": {
    "multifield-classifier-name": "global_mfc",
    "remark-map-name": "subs-remarking-residential"
  }
}
```

#### Remark-map to Instance Mapping

Use the following CLI syntax to configure the remark map for an instance.

**set forwarding-options class-of-service instance** <instance-name> **remark-map-name** <remark-map-name>

### Command arguments

| Attribute | Description |
|---|---|
| `<instance-name> | Specifies the name of the instance.` |
| <remark-map-name> | Specifies the name of the remark map. |

The following example configures the remark map for the instance `default`.

```
set forwarding-options class-of-service instance default remark-map-name subs-remarking-residential
commit
```

The following example shows the remark map configuration for the instance `default`.

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service instance default
```

```
{
  "rtbrick-config:instance": [
    {
      "name": "default",
      "remark-map-name": "subs-remarking-residential"
    }
  ]
}
```

### Configuring QoS L2 and L3 Logical Interfaces

#### QoS Configuration

To support QoS on L2 logical interface, the class-of-service profile configuration is introduced. If a L2 logical interface is part of an interface-set, logical-interface scheduler/shaper must have parent-scheduler as interface-set-scheduler/shaper.

```
set interface ifp-0/0/16 unit 201
set interface ifp-0/0/16 unit 201 interface-type l2vpn-vpws
set interface ifp-0/0/16 unit 201 instance evpn-vpws-vrf2
set interface ifp-0/0/16 unit 201 vlan 200
set interface ifp-0/0/16 unit 201 inner-vlan 201
set interface ifp-0/0/16 unit 201 class-of-service qos_profile_service_1
set interface ifp-0/0/16 unit 202
set interface ifp-0/0/16 unit 202 interface-type l2vpn-vpws
set interface ifp-0/0/16 unit 202 instance evpn-vpws-vrf2
set interface ifp-0/0/16 unit 202 vlan 200
set interface ifp-0/0/16 unit 202 inner-vlan 202
set interface ifp-0/0/16 unit 202 class-of-service qos_profile_service_2
set interface ifp-0/0/16 unit 203
set interface ifp-0/0/16 unit 203 interface-type l2vpn-vpws
set interface ifp-0/0/16 unit 203 instance evpn-vpws-vrf2
set interface ifp-0/0/16 unit 203 vlan 200
set interface ifp-0/0/16 unit 203 inner-vlan 203
set interface ifp-0/0/16 unit 203 class-of-service qos_profile_service_3
```

#### Interface-set Configuration

```
set interface-set service-group-1 interface ifp-0/0/16
set interface-set service-group-1 interface ifp-0/0/16 unit 201
set interface-set service-group-1 interface ifp-0/0/16 unit 202
set interface-set service-group-1 interface ifp-0/0/16 unit 203
```

#### Combining Multiple Logical Interfaces with an Added QoS Hierarchy

#### Configuring QoS Profile for a Logical Interface or Service

```
set forwarding-options class-of-service profile qos_profile_service_1
set forwarding-options class-of-service profile qos_profile_service_1 classifier-name upstream_pbit_classifer
```

```
set forwarding-options class-of-service profile qos_profile_service_1 class-queue-map-name
business_service_class_queue_map
set forwarding-options class-of-service profile qos_profile_service_1 class-policer-map-name
business_service_class_policer_map
set forwarding-options class-of-service profile qos_profile_service_1 policer-name business_service_policer_1
set forwarding-options class-of-service profile qos_profile_service_1 scheduler-map-name business_service_scheduler_map_1
```

#### Configuring QoS Profile for a Logical Interface Set

```
set forwarding-options class-of-service profile qos_profile_service_group
set forwarding-options class-of-service profile qos_profile_service_group scheduler-map-name
business_service_group_scheduler_map
```

#### Enabling QoS Profile for a Logical Interface Set

```
set interface-set service-group-1
set interface-set service-group-1 class-of-service qos_profile_service_group
set forwarding-options class-of-service profile service-group-profile
set forwarding-options class-of-service profile service-group-profile class-queue-map-name subs-4queues
set forwarding-options class-of-service profile service-group-profile class-policer-map-name policer-map-residential
set forwarding-options class-of-service profile service-group-profile policer-name policer-residential
set forwarding-options class-of-service profile service-group-profile scheduler-map-name
business_service_group_scheduler_map
```

The following CLIs are used to configure scheduler port connection to link schedulers across different scheduler maps:

QoS Scheduler-map to connect logical interface scheduler with interface-set scheduler:

```
set forwarding-options class-of-service scheduler-map schedmap_service_1
set forwarding-options class-of-service scheduler-map schedmap_service_1  scheduler-name sch_service_1
set forwarding-options class-of-service scheduler-map schedmap_service_1  scheduler-name sch_service_1 port-connection
scheduler-map-to-scheduler-map
set forwarding-options class-of-service scheduler-map schedmap_service_1   scheduler-name business_service_scheduler_1
parent-scheduler-name business_service_group_scheduler
```

#### QoS Scheduler-map to Connect an interface-set scheduler with a pon scheduler

```
set forwarding-options class-of-service scheduler-map business_service_group_scheduler_map
set forwarding-options class-of-service scheduler-map business_service_group_scheduler_map scheduler-name
business_service_group_scheduler
set forwarding-options class-of-service scheduler-map business_service_group_scheduler scheduler-name
business_service_group_scheduler parent-scheduler-name pon0
set forwarding-options class-of-service scheduler-map business_service_group_scheduler_map scheduler-name
business_service_group_scheduler port-connection scheduler_map_to_scheduler_map
```

### QoS Global and Instance Configurations

The figure below shows the dependencies for per instance or global classifier and remark-map configurations.

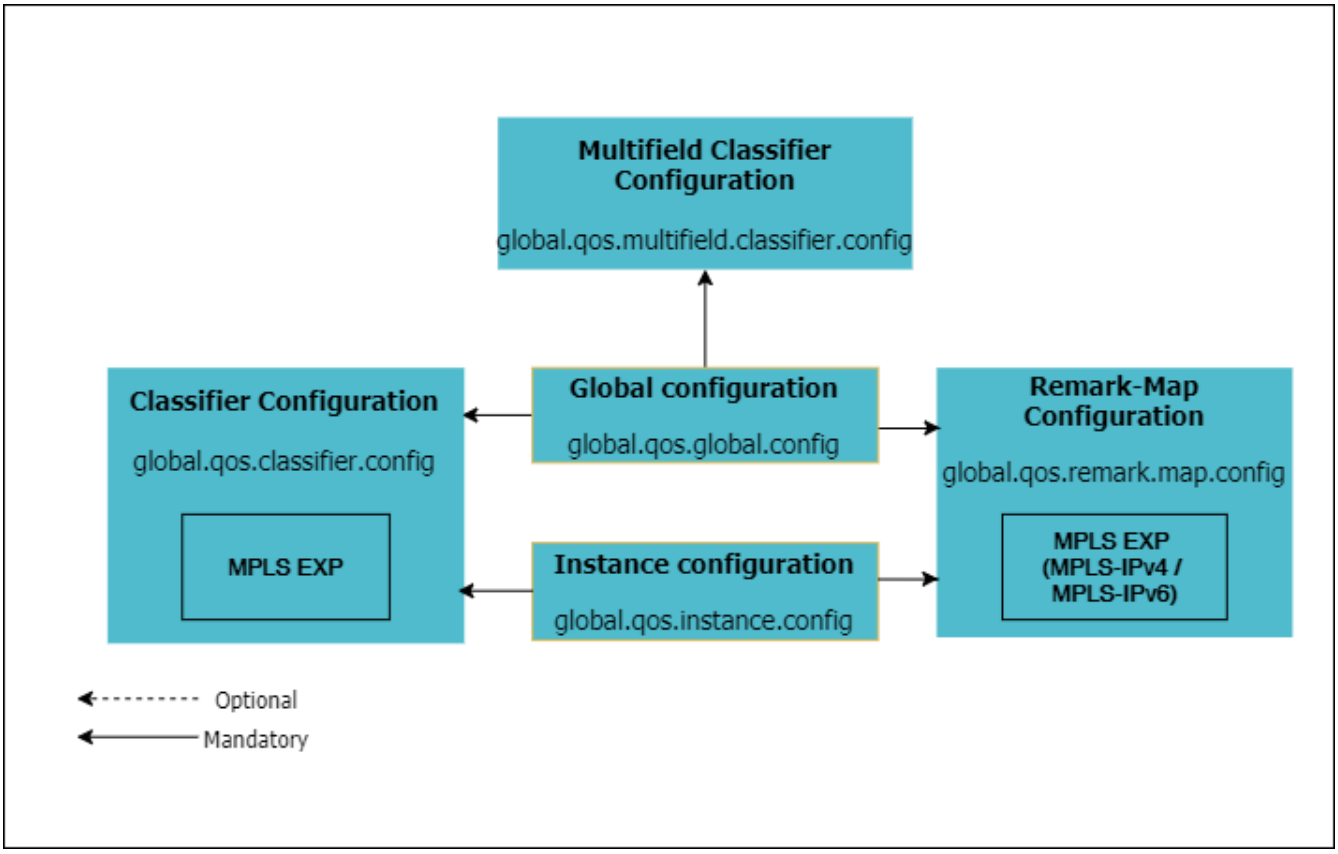



*Figure 144. QoS Global and Instance Configurations*

On the Q2C platforms, configuring an instance-level remark map with the MPLS EXP type is not supported.

The following example configures the BA classifier `subs-exp-class` for traffic `exp` with match code point `2` classified as class `class-1`.

```
set forwarding-options class-of-service classifier subs-exp-class match-type exp codepoint 2 class class-1
```

#### BA Classifier to Global Mapping

The MPLS classifiers can be applied globally using global configuration.

**set forwarding-options class-of-service global classifier-name** <classifier-name>

| Attribute | Description |
|---|---|
| `<classifier-name>` | Specifies the classifier name |

The following example shows how to configure the `subs-exp-class` BA Classifier for global mapping.

```
set forwarding-options class-of-service global classifier-name subs-exp-class
```

The configuration for global mapping to the BA Classifier is shown in the following example.

```
supervisor@rtbrick>LEAF01: cfg> show config  forwarding-options class-of-service global classifier-name
{
  "rtbrick-config:classifier-name": "subs-exp-class"
}
```

#### Single Queue To Global Mapping

The Single-Queue can be enabled with global configuration.

> For the system to initialize with a single queue in a Queue-Group, a system reboot is required.

**set forwarding-options class-of-service global queue-group-profile** single-queue

The following example shows how to enable single-queue in a Queue-Group.

```
set forwarding-options class-of-service global queue-group-profile single-queue
commit
```

The configuration for a single queue in Queue-Group is shown in the following example.

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service global
{
  "rtbrick-config:global": {
    "queue-group-profile": "single-queue"
  }
}
```

#### Remark-Map To Global Mapping

Use the following CLI syntax to configure the remark map for a global profile.

**set forwarding-options class-of-service global remark-map-name** <remark-map-name>

### Command arguments

| Attribute | Description |
|---|---|
| `<remark-map-name>` | Specifies the name of the remark map. |

The following example configures the remark map for a global profile:

```
set forwarding-options class-of-service global remark-map-name subs-remarking-residential
commit
```

#### Multifield Classifier (MFC) to Global Mapping

Use the following CLI syntax to configure the MF Classifier to Global Mapping:

**set forwarding-options class-of-service global multifield-classifier-name** <multifield-classifier-name>

| Attribute | Description |
|---|---|
| `<multifield-classifier-name>` | Specifies the name of the multifield classifier |

The following example configures the multifield Classifier to Global Mapping:

```
set forwarding-options class-of-service global multifield-classifier-name global_mfc
commit
```

The following example shows the multifield classifier to Global Mapping configuration:

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service global
{
  "rtbrick-config:global": {
    "multifield-classifier-name": "global_mfc"
  }
}
```

#### BA Classifier to Instance Mapping

The MPLS classifiers can be applied at an instance level using instance configuration.

**set forwarding-options class-of-service instance** <instance> **classifier-name** <classifier-name>

| Attribute | Description |
|---|---|
| `<instance>` | Specifies the instance name |
| `<classifier-name>` | Specifies the classifier name |

The following example shows how to configure the `subs-exp-class` BA Classifier for the instance `default`.

```
set forwarding-options class-of-service instance default classifier-name subs-exp-class
commit
```

The configuration for instance mapping to the BA Classifier is shown in the following example.

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service instance default
{
  "rtbrick-config:instance": [
    {
      "name": "default",
      "classifier-name": "subs-exp-class"
    }
  ]
}
```

#### Remark-Map To Instance Mapping

Use the following CLI syntax to configure the remark map for an instance.

**set forwarding-options class-of-service instance <instance-name> remark-map-name** <remark-map-name>

### Command arguments

| Attribute | Description |
|---|---|
| `<instance-name>` | Specifies the name of the instance. |
| `<remark-map-name>` | Specifies the name of the remark map. |

The following example configures the remark map for the instance `default`:

```
set forwarding-options class-of-service instance default remark-map-name subs-remarking-residential
commit
```

The configuration for the remark map for an instance is shown in the following example.

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service instance default
{
  "rtbrick-config:instance": [
    {
      "name": "default",
      "remark-map-name": "subs-remarking-residential"
    }
  ]
}
```

#### Configuring Control Plane QoS Per IFL

Control plane packet marking is supported on a per-IFL basis for the following protocols: BGP, OSPF, LDP, IS-IS, DHCP relay, RADIUS, L2TP, ARP/ND, and PIM. Additionally, IGMP control plane packets can now be marked on a per-IFP basis.

**Syntax:**

**set forwarding-options class-of-service control-plane-qos outbound-marking protocol** <protocol-name> **tos codepoint** <value>

**set forwarding-options class-of-service control-plane-qos outbound-marking protocol** <protocol-name> **tos interface** <interface-name> **codepoint** <value>

| Attribute | Description |
|---|---|
| `<protocol-name>` | Specifies the name of the protocol. |
| `<interface-name>` | Specifies the name of the IFL. |
| `codepoint <value>` | Specify the codepoint value. |

Example configuration for configuring control plane QoS per IFL:

```
set forwarding-options class-of-service control-plane-qos outbound-marking protocol
bgp tos codepoint 20
set forwarding-options class-of-service control-plane-qos outbound-marking protocol
bgp tos ifl ifl-0/0/0/1001
set forwarding-options class-of-service control-plane-qos outbound-marking protocol
bgp tos ifl ifl-0/0/0/3003 codepoint 19
set forwarding-options class-of-service control-plane-qos outbound-marking protocol
bgp tos ifl ifl-0/0/0/3003
set forwarding-options class-of-service control-plane-qos outbound-marking protocol
bgp tos ifl ifl-0/0/0/3003 codepoint 19
```

```
{
    "protocol": "bgp",
    "remark-type": "tos",
    "codepoint": 20,
    "ifl": [
        {
            "ifl": "ifl-0/0/0/1001",
            "codepoint": 19
        },
        {
            "ifl": "ifl-0/0/0/3003",
            "codepoint": 19
        }
    ]
}
```

#### Control Plane QoS p-bit Support for OAM

RBFS supports control plane QoS p-bit remarking for OAM packets at both the protocol level and

the L2-all level.

**Syntax:**

**set forwarding-options class-of-service control-plane-qos outbound-marking protocol cfm p-bit codepoint** <value>

| Attribute | Description |
|---|---|
| `codepoint <value>` | Specify the codepoint value. |

Example Command:

```
set forwarding-options class-of-service control-plane-qos outbound-marking protocol
cfm p-bit codepoint 4
```

Example:

```
"rtbrick-config:forwarding-options": {
      "class-of-service": {
        "control-plane-qos": {
          "outbound-marking": {
            "protocol": [
              {
                "protocol": "cfm",
                "remark-type": "p-bit",
                "codepoint": 4
              }
            ]
          }
        }
      }
    },
```

#### Control Plane Marking (ToS) for MPLS Ping and Traceroute

This feature allows you to set the ToS value in the header of MPLS OAM ping and traceroute packets generated or received by the control plane. It enables the prioritization of MPLS OAM traffic, thereby giving higher priority to diagnostic traffic.

**Syntax:**

**set forwarding-options class-of-service control-plane-qos outbound-marking protocol mpls-oam tos codepoint** <value>

| Attribute | Description |
|---|---|
| `codepoint <value>` | Specify the codepoint value. |

Example Command:

```
set forwarding-options class-of-service control-plane-qos outbound-marking protocol
mpls-oam tos codepoint 180
```

Example:

```
supervisor@rtbrick.net: cfg> show config forwarding-options class-of-service control-
plane-qos outbound-marking protocol mpls-oam
{
  "rtbrick-config:protocol": [
    {
      "protocol": "mpls-oam",
      "remark-type": "tos",
      "codepoint": 180
    }
  ]
}
```

#### Configuring QoS for L2TP LCP Control Plane Packets

RBFS supports for prioritizing L2TP LCP control packets using Layer 3 IPv4 (L3v4) ACLs. It enables the identification of LCP control plane packets and allows for Quality of Service (QoS) marking.

**Syntax:**

**set forwarding-options acl l3v4 rule l2tp_match_lcp ordinal** <number> <attribute> <value>

| Attribute | Description |
|---|---|
| `ordinal <ordinal-value>` | Specifies the ordinal that is used for traffic rule referencing. |
| `match <direction>` | Specifies the direction of L3V4 traffic. |
| `match l2tp-ppp-lcp true` | Specify 'true' to enable the rule to match the L2TP LCP traffic. |
| `action forward-class <class>` | class-0, class-1, class-2, class-3, class-4, class-5, class-6, class-7 |
| `action remark-codepoint <remark-codepoint>` | Specifies the remark-map codepoint value. |
| `action statistics true` | Specify 'true' to generate statistics for the ACL. |
| `priority <value>` | Specify the priority value. |

The following configuration defines a Layer 3 IPv4 ACL rule to match incoming L2TP PPP LCP packets with specific actions for QoS marking. The name of the L3 IPv4 ACL rule is specified as "l2tp_match_lcp."

The ordinal value, set to 1, indicates the order of execution when there are multiple match-action entries. The match direction is defined as "ingress", indicating that this rule applies to incoming traffic. The match for "l2tp-ppp-lcp" is set to 'true'. It indicates it is applicable to only L2TP PPP LCP control packets.

The action "forward-class" assigns the packet to forwarding class 3 for higher priority. The action "remark-codepoint" is set to the value '100' that help to classify the packets for special QoS treatment. Also, the "action statistics" is enabled by setting it to 'true'. It enables the generation of statistics for this rule. The priority is set to '100', assigning priority level 100 to this rule.

```
set forwarding-options acl l3v4 rule l2tp_match_lcp
set forwarding-options acl l3v4 rule l2tp_match_lcp ordinal 1
set forwarding-options acl l3v4 rule l2tp_match_lcp ordinal 1 match direction ingress
set forwarding-options acl l3v4 rule l2tp_match_lcp ordinal 1 match l2tp-ppp-lcp true
set forwarding-options acl l3v4 rule l2tp_match_lcp ordinal 1 action forward-class
class-3
set forwarding-options acl l3v4 rule l2tp_match_lcp ordinal 1 action remark-codepoint
100
set forwarding-options acl l3v4 rule l2tp_match_lcp ordinal 1 action statistics true
set forwarding-options acl l3v4 rule l2tp_match_lcp ordinal 1 priority 100
```

```
{
  "rtbrick-config:acl": {
    "l3v4": {
      "rule": [
        {
          "rule-name": "l2tp_match_lcp",
          "ordinal": [
            {
              "ordinal-value": 1,
              "match": {
                "direction": "ingress",
                "l2tp-ppp-lcp": "true"
              },
              "action": {
                "forward-class": "class-3",
                "remark-codepoint": 100,
                "statistics": "true"
              },
              "priority": 100
            }
          ]
        }
      ]
    }
```

```
  }
}
```

**DSCP Remarking with ECN Transparency**

This feature allows remarking of DSCP (Differentiated Services Code Point) values in IP packets while preserving the ECN (Explicit Congestion Notification) bits. With this feature, RBFS can modify only the DSCP portion of the IP header while keeping the ECN field unchanged.

- To ensure ECN transparency, you must configure the MFC with the remark-dscp-codepoint option. Also, the traffic must match and hit the MFC ACL.
- You cannot configure the remark-dscp-codepoint action when match condition is either ipv4-dscp or ipv4-tos or ipv6-tc.
- Remark code-point (remark-codepoint) and Remark DSCP code-point (remark-dscp-codepoint) are mutually exclusive configurations and cannot be used together.

The following configuration example defines a traffic filter condition and what to perform with matched traffic. It defines when a packet's destination IP is within 132.1.1.0/24 range and the destination L4 port is 65057, then send it using forwarding class 'class-1' and rewrite its DSCP marking to 60.

```
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 101 match source-ipv4-prefix 132.1.1.1/24
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 101 match source-l4-port
65057
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 101 action
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 101 action forward-class class-1
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 101 action remark-dscp-codepoint 60
```

```
{
  "rtbrick-config:ordinal": [
    {
      "ordinal-value": 1,
      "match": {
        "destination-ipv4-prefix": "132.1.1.1/24",
        "destination-l4-port": 65057
      },
      "action": {
        "forward-class": "class-1",
        "remark-dscp-codepoint": 60
      }
    }
```

```
  ]
}
```

### 6.7.3. HQoS Show Running-Configuration

To display the running configuration, use the **show config** command.

#### Syntax

**show config**

#### Example

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options
    "rtbrick-config:forwarding-options": {
      "mirror": [
        {
          "name": "m1",
          "destination": {
            "interface": "cpu-0/0/200"
          },
          "source": {
            "direction": "ingress",
            "interface": "ifp-0/1/30"
          }
        }
      ],
      "class-of-service": {
        "classifier": [
          {
            "classifier-name": "subs-pbit-class",
            "match-type": [
              {
                "match-type": "ieee-802.1",
                "codepoint": [
                  {
                    "codepoint": 1,
                    "class": "class-0",
                    "remark-codepoint": 7
                  },
                  {
                    "codepoint": 2,
                    "class": "class-1",
                    "remark-codepoint": 7
                  },
                  {
                    "codepoint": 3,
                    "class": "class-2",
                    "remark-codepoint": 7
                  },
```

```
                {
                  "codepoint": 4,
                  "class": "class-3",
                  "remark-codepoint": 7
                }
              ]
            }
          ]
        }
      ],
      "class-policer-map": [
        {
          "class-policer-map-name": "policer-map-residential",
          "class": [
            {
              "class": "class-0",
              "policer-level": "level-1"
            },
            {
              "class": "class-1",
              "policer-level": "level-2"
            },
            {
              "class": "class-2",
              "policer-level": "level-3"
            },
            {
              "class": "class-3",
              "policer-level": "level-4"
            }
          ]
        }
      ],
      "class-queue-map": [
        {
          "class-queue-map-name": "subs-4queues",
          "class": [
            {
              "class-type": "class-0",
              "queue-name": "BE_SUBS"
            },
            {
              "class-type": "class-1",
              "queue-name": "LD_SUBS"
            },
            {
              "class-type": "class-2",
              "queue-name": "LL_SUBS"
            },
            {
              "class-type": "class-3",
```

```
                "queue-name": "VO_SUBS"
              }
            ]
          }
        ],
        "global": {
          "multifield-classifier-name": "global_mfc"
        },
        "policer": [
          {
            "policer-name": "policer-residential",
            "level1-rates": {
              "cir": 2000,
              "cbs": 1000,
              "pir": 2500,
              "pbs": 1000
            },
            "level2-rates": {
              "cir": 3000,
              "cbs": 1000,
              "pir": 3500,
              "pbs": 1000
            },
            "level3-rates": {
              "cir": 4000,
              "cbs": 1000,
              "pir": 4500,
              "pbs": 1000
            },
            "level4-rates": {
              "cir": 1000,
              "cbs": 1000,
              "pir": 1500,
              "pbs": 1000
            },
            "levels": 4,
            "type": "two-rate-three-color"
          }
        ],
        "profile": [
          {
            "profile-name": "residential",
            "classifier-name": "subs-pbit-class",
            "class-queue-map-name": "subs-4queues",
            "remark-map-name": "subs-remarking-residential",
            "class-policer-map-name": "policer-map-residential",
            "policer-name": "policer-residential",
            "scheduler-map-name": "subs-4queues-residential"
          }
        ],
        "queue": [
```

```
          {
            "queue-name": "BE_SUBS",
            "queue-size": 375000
          },
          {
            "queue-name": "LD_SUBS",
            "queue-size": 625000
          },
          {
            "queue-name": "LL_SUBS",
            "queue-size": 625000
          },
          {
            "queue-name": "VO_SUBS",
            "queue-size": 156250,
            "shaper-name": "shaper_VO"
          }
        ],
        "queue-group": [
          {
            "queue-group-name": "subs-4queues",
            "queue-numbers": 4
          }
        ],
        "remark-map": [
          {
            "remark-map-name": "subs-remarking-residential",
            "remark-type": [
              {
                "remark-type": "ieee-802.1",
                "match-codepoint": [
                  {
                    "match-codepoint": 128,
                    "color": [
                      {
                        "color": "all",
                        "remark-codepoint": 6
                      }
                    ]
                  },
                  {
                    "match-codepoint": 160,
                    "color": [
                      {
                        "color": "all",
                        "remark-codepoint": 6
                      }
                    ]
                  },
                  {
                    "match-codepoint": 192,
```

```
                "color": [
                  {
                    "color": "all",
                    "remark-codepoint": 6
                  }
                ]
              },
              {
                "match-codepoint": 224,
                "color": [
                  {
                    "color": "all",
                    "remark-codepoint": 6
                  }
                ]
              }
            ]
          }
        ]
      }
    ],
    "scheduler": [
      {
        "scheduler-name": "pon0",
        "shaper-name": "gpon-shaper",
        "type": "fair_queueing"
      },
      {
        "scheduler-name": "subs-4queues",
        "shaper-name": "shaper_session",
        "type": "strict_priority",
        "composite": "false"
      }
    ],
    "scheduler-map": [
      {
        "scheduler-map-name": "schedmap-olt",
        "scheduler-name": [
          {
            "name": "pon0",
            "port-connection": "scheduler_to_port"
          }
        ]
      },
      {
        "scheduler-map-name": "subs-4queues-residential",
        "queue-group-name": [
          {
            "group-name": "subs-4queues",
            "queue-name": [
              {
```

```
                "name": "BE_SUBS",
                "parent-flow": "high-flow",
                "parent-scheduler-name": "subs-4queues",
                "connection-point": "strict_priority_3"
              },
              {
                "name": "LD_SUBS",
                "parent-flow": "high-flow",
                "parent-scheduler-name": "subs-4queues",
                "connection-point": "strict_priority_1"
              },
              {
                "name": "LL_SUBS",
                "parent-flow": "high-flow",
                "parent-scheduler-name": "subs-4queues",
                "connection-point": "strict_priority_2"
              },
              {
                "name": "VO_SUBS",
                "parent-flow": "high-flow",
                "parent-scheduler-name": "subs-4queues",
                "connection-point": "strict_priority_0"
              }
            ]
          }
        ],
        "scheduler-name": [
          {
            "name": "subs-4queues",
            "port-connection": "scheduler_to_port"
          }
        ]
      }
    ],
    "shaper": [
      {
        "shaper-name": "shaper_VO",
        "shaping-rate-high": 2000,
        "shaping-rate-low": 0
      },
      {
        "shaper-name": "shaper_session",
        "shaping-rate-high": 10000,
        "shaping-rate-low": 100
      },
      {
        "shaper-name": "gpon-shaper",
        "shaping-rate-high": 2488000,
        "shaping-rate-low": 32000
      }
    ],
```

```
        "multifield-classifier": {
          "acl": {
            "l3v4": {
              "rule": [
                {
                  "rule-name": "global_mfc",
                  "ordinal": [
                    {
                      "ordinal-value": 1001,
                      "match": {
                        "ipv4-tos": 128,
                        "source-ipv4-prefix": "192.0.2.2/32"
                      },
                      "action": {
                        "forward-class": "class-0"
                      }
                    },
                    {
                      "ordinal-value": 1002,
                      "match": {
                        "ipv4-tos": 160,
                        "source-ipv4-prefix": "192.0.2.2/32"
                      },
                      "action": {
                        "forward-class": "class-1"
                      }
                    },
                    {
                      "ordinal-value": 1003,
                      "match": {
                        "ipv4-tos": 192,
                        "source-ipv4-prefix": "192.0.2.2/32"
                      },
                      "action": {
                        "forward-class": "class-2"
                      }
                    },
                    {
                      "ordinal-value": 1004,
                      "match": {
                        "ipv4-tos": 224,
                        "source-ipv4-prefix": "192.0.2.2/32"
                      },
                      "action": {
                        "forward-class": "class-3"
                      }
                    }
                  ]
                }
              ]
            }
```

```
          }
        }
      }
    }
```

## 6.7.4. HQoS Operational Commands

### HQoS Show Commands

The HQoS show commands provide detailed information about the HQoS operation.

#### show qos classifier

**Syntax:**

**show qos classifier** <option>

| Option | Description |
|---|---|
| `all` | Displays a summary of all the configured classifiers. |
| `interface <interface_name>` | Displays only the classifiers applied to a specific interface. |
| `type <key_type>` | Key type indicates the interface category, for example, whether its subscriber interface, an L3 IFL, a physical interface and so on. |

The following example displays information for the specified classifier.

```
supervisor@rtbrick: op> show qos classifier all interface ppp-0/1/23/72339069014638634
Interface: ppp-0/1/23/72339069014638634, Classifier: SUBS_PBIT_CLASSIFIER, Type: Subscriber
  Match Type           Codepoint  Class     Remark Codepoint Color
  ieee-802.1                   3  class-3                160 -
  ieee-802.1                   2  class-2                140 -
  ieee-802.1                   0  class-0                100 -
  ieee-802.1                   1  class-1                120 -
```

The following example displays information for the specified key type.

```
supervisor@rtbrick: op> show qos classifier all type Global
Classifier: SUBS_EXP_CLASSIFIER, Type: Global
  Match Type           Codepoint  Class     Remark Codepoint Color
  exp                          0  class-0                  - -
  exp                          1  class-1                  - -
  exp                          2  class-2                  - -
  exp                          3  class-3                  - -
```

#### show qos interface

**Syntax:**

**show qos interface** <option>

| Option | Description |
|---|---|
| - | Without any option, this command displays a summary of all the interfaces. |
| <interface-name> | Displays QoS classifier information for the specified interface. |

The following example displays information for the specified interface.

```
supervisor@rtbrick: op> show qos interface ifl-0/1/24/200
Interface                          Profile
ifl-0/1/24/200                     SUBSCRIBER_QOS_PROFILE
```

#### show qos policer

**Syntax:**

**show qos policer** <option>

| Option | Description |
|---|---|
| all | Displays a summary of all the QoS policers. |
| <policer-name> | Displays QoS policer information. |
| counter | Displays policer counter information. |
| <interface-name> | Displays policers applied to a specific interface. |
| type <key_type> | Key type indicates the interface category, for example, whether its subscriber interface, an L3 IFL, a physical interface and so on. |
| class-to-policer-map | Displays class-to-policer-map information. |

The following example displays a summary of all the QoS policers.

```
supervisor@rtbrick: op> show qos policer all | head -n 20
Key Name: _DEFAULT_POLICER_10_MB, Key Type: Acl, Policer: _DEFAULT_POLICER_10_MB
Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(Kbits)     PBS(Kbits)     Max CIR(Kbps)  Max PIR(Kbps)
  1        10000          10000          33000          33000          -              -
  2        -              -              -              -              -              -
  3        -              -              -              -              -              -
  4        -              -              -              -              -              -
Key Name: _DEFAULT_POLICER_1_MB, Key Type: Acl, Policer: _DEFAULT_POLICER_1_MB
```

```
Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(Kbits)     PBS(Kbits)     Max CIR(Kbps)  Max PIR(Kbps)
  1        1000           1000           33000          33000          -              -
  2        -              -              -              -              -              -
  3        -              -              -              -              -              -
  4        -              -              -              -              -              -
Key Name: ppp-0/1/23/72339069014638637, Key Type: Subscriber-Ifl, Policer: SUBS_POLICER
Type: two-rate-three-color, Levels: 4, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(Kbits)     PBS(Kbits)     Max CIR(Kbps)  Max PIR(Kbps)
  1        2000           2100           200            210            -              -
  2        3000           3100           300            310            -              -
  3        1000           1100           100            110            -              -
```

The following example displays a summary of all the QoS policers for a specified interface.

```
supervisor@rtbrick: op> show qos policer all interface ppp-0/1/23/72339069014638634
Key Name: ppp-0/1/23/72339069014638634, Key Type: Subscriber-Ifl, Policer:
SUBS_POLICER
Type: two-rate-three-color, Levels: 4, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(Kbits)     PBS(Kbits)     Max CIR(Kbps)
Max PIR(Kbps)
  1        2000           2100           200            210            -
-
  2        3000           3100           300            310            -
-
  3        1000           1100           100            110            -
-
  4        4000           4100           400            410            -
-
```

The following example displays a summary of all the QoS policers for a specified key type.

```
supervisor@rtbrick: op> show qos policer all type NAT
Key Name: _DEFAULT_NAT_POLICER_50_MB, Key Type: NAT, Policer: _DEFAULT_NAT_POLICER_50_MB
Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(Kbits)     PBS(Kbits)     Max CIR(Kbps)  Max PIR(Kbps)
  1        50000          50000          33000          33000          -              -
  2        -              -              -              -              -              -
  3        -              -              -              -              -              -
  4        -              -              -              -              -              -
```

The following example displays counter information for the specified counter.

```
supervisor@rtbrick>LEAF01: op> show qos policer counter lag-1
Interface                        Level  Units      Total            Received         Dropped
lag-1                            1      Packets    15773            15773            0
                                        Bytes      2555226          2555226          0
lag-1                            2      Packets    15778            15778            0
                                        Bytes      2556036          2556036          0
lag-1                            3      Packets    15775            15775            0
```

```
                                        Bytes      2555550          2555550          0
lag-1                            4      Packets    23661            23661            0
                                        Bytes      3423166          3423166          0
lag-1-egress                     1      Packets    7889             7889             0
                                        Bytes      1420020          1420020          0
lag-1-egress                     2      Packets    7889             7889             0
                                        Bytes      1420020          1420020          0
lag-1-egress                     3      Packets    7889             7889             0
                                        Bytes      1420020          1420020          0
lag-1-egress                     4      Packets    49177            49177            0
                                        Bytes      8288974          8288974          0
```

#### show qos profile

**Syntax:**

**show qos profile** <option>

| Option | Description |
|---|---|
| - | Without any option, this command displays a summary of all the QoS profiles. |
| <profile-name> | Displays QoS profile information. |

The following example displays a summary of a specific QoS profile.

```
supervisor@rtbrick: op> sho qos profile SUBSCRIBER_QOS_PROFILE
Profile: SUBSCRIBER_QOS_PROFILE
  Classifier: SUBS_PBIT_CLASSIFIER
  Policer: SUBS_POLICER
  Egress Policer: -
  Scheduler map: SUBS_QUEUES_4_SERVICES
  Class queue map: SUBS_QUEUES_4
  Remark map: SUBS_REMARKING_EGRESS
  Class policer map: SUBS_CLASS_POLICER_MAP
  Egress Class policer map: -
  Mulifield classifier: -
```

#### show qos queue

**Syntax:**

**show qos queue** <option>

| Option | Description |
|---|---|
| all | Displays all configured QoS queues and their operational state. |

| Option | Description |
|---|---|
| `<interface_name>` | Displays queues assigned to a specific interface. |
| `<interface_name> <queue_name>` | Displays details for a specific queue on a specified interface. |
| `class-to-queue-map` | Displays queue class-to-queue-map information. |
| `counter` | Displays QoS queue counter information. |
| `counter <interface-name>` | Displays QoS queue counter information for the specified interface. |
| `type <key_type>` | Key type indicates the interface category, for example, whether its subscriber interface, an L3 IFL, a physical interface and so on. |

The following example displays queue information for the specified interface.

```
supervisor@rtbrick: op> show qos queue all interface ppp-0/1/23/72339069014638634
Interface: ppp-0/1/23/72339069014638634, Type: Subscriber
  Queue                      Queue Size      Min Thres      Max Thres      Drop Prob  Shaper
  QUEUE_0                        375000              -              -              -  -
  QUEUE_1                        625000              -              -              -  -
  QUEUE_2                        625000              -              -              -  -
  QUEUE_3                        156250              -              -              -  SUBS_SHAPER_QUEUE_3
```

The following example displays queue information for a specified key type.

```
supervisor@rtbrick: op> show qos queue all type Subscriber-Ifl | head -n 5
Interface: ppp-0/1/23/72339069014638637, Type: Subscriber
  Queue                      Queue Size      Min Thres      Max Thres      Drop Prob  Shaper
  QUEUE_0                        375000              -              -              -  -
  QUEUE_1                        625000              -              -              -  -
  QUEUE_2                        625000              -              -              -  -
```

The following example displays queue counter information.

```
supervisor@rtbrick: op> show qos queue counter | head -n 20
Interface                    Queue Group            Queue                  Class    Units      Received
Queued              Dropped
ifl-0/1/24/200               SUBS_QUEUES_4          QUEUE_0                0        Packets    15848              3180
12668
                                                                                    Bytes      16133264
3237240             12896024
ifl-0/1/24/200               SUBS_QUEUES_4          QUEUE_1                1        Packets    15849
15849               0
                                                                                    Bytes      16134282
16134282            0
ifl-0/1/24/200               SUBS_QUEUES_4          QUEUE_2                2        Packets    15849
11185               4664
                                                                                    Bytes      16134282
11386330            4747952
ifl-0/1/24/200               SUBS_QUEUES_4          QUEUE_3                3        Packets    15848              4776
11072
                                                                                    Bytes      16133264
4861968             11271296
```

```
ifp-0/1/23                    _DEFAULT_IFP_QUEUE_G   _DEFAULT_IFP_QUEUE_0   0        Packets    392925
392925             0
                                                                                      Bytes      399923880
399923880          0
ifp-0/1/23                    _DEFAULT_IFP_QUEUE_G   _DEFAULT_IFP_QUEUE_1   1        Packets    0                  0
0
                                                                                      Bytes      0                  0
0
ifp-0/1/23                    _DEFAULT_IFP_QUEUE_G   _DEFAULT_IFP_QUEUE_2   2        Packets    0                  0
0
                                                                                      Bytes      0                  0
0
ifp-0/1/23                    _DEFAULT_IFP_QUEUE_G   _DEFAULT_IFP_QUEUE_3   3        Packets    0                  0
0
                                                                                      Bytes      0                  0
0
ifp-0/1/23                    _DEFAULT_IFP_QUEUE_G   _DEFAULT_IFP_QUEUE_4   4        Packets    0                  0
0
                                                                                      Bytes      0                  0
0
ifp-0/1/23                    _DEFAULT_IFP_QUEUE_G   _DEFAULT_IFP_QUEUE_5   5        Packets    0                  0
0
```

The following example displays queue counter information for the specified interface.

```
supervisor@rtbrick: op> show qos queue counter ifl-0/1/24/200
Interface                     Queue Group            Queue                  Class    Units      Received
Queued             Dropped
ifl-0/1/24/200                SUBS_QUEUES_4          QUEUE_0                0        Packets    15848              3180
12668
                                                                                      Bytes      16133264
3237240            12896024
ifl-0/1/24/200                SUBS_QUEUES_4          QUEUE_1                1        Packets    15849
15849              0
                                                                                      Bytes      16134282
16134282           0
ifl-0/1/24/200                SUBS_QUEUES_4          QUEUE_2                2        Packets    15849
11185              4664
                                                                                      Bytes      16134282
11386330           4747952
ifl-0/1/24/200                SUBS_QUEUES_4          QUEUE_3                3        Packets    15848              4776
11072
                                                                                      Bytes      16133264
4861968            11271296
```

#### show qos scheduler

### Syntax:

**show qos scheduler** <option>

| Option | Description |
|---|---|
| `all` | Displays all scheduler profiles configured. |
| `<scheduler-name>` | Displays information for the specified scheduler. |
| `type <key_type>` | Displays schedulers for the specified type. |

The following example displays a summary of all the schedulers.

```
supervisor@rtbrick: op> show qos scheduler all
Interface Name                 Type                      Scheduler Name  Scheduler Map Name            Shaper Name
Composite
ppp-0/1/23/72339069014638637   Subscriber-Ifl            SUBS_QUEUES_4   SUBS_QUEUES_4_SERVICES
SUBS_SESSION_SHAPER False
ifp-0/1/46                     Physical-Interface-Sch... SUBS_PON_0      PORT_SCHEDULER                -
False
ppp-0/1/23/72339069014638634   Subscriber-Ifl            SUBS_QUEUES_4   SUBS_QUEUES_4_SERVICES
SUBS_SESSION_SHAPER False
ppp-0/1/23/72339069014638638   Subscriber-Ifl            SUBS_QUEUES_4   SUBS_QUEUES_4_SERVICES
SUBS_SESSION_SHAPER False
ifl-0/1/24/200                 L3-Ifl-Interface          SUBS_QUEUES_4   SUBS_QUEUES_4_SERVICES
SUBS_SESSION_SHAPER False
ppp-0/1/23/72339069014638636   Subscriber-Ifl            SUBS_QUEUES_4   SUBS_QUEUES_4_SERVICES
SUBS_SESSION_SHAPER False
ppp-0/1/23/72339069014638635   Subscriber-Ifl            SUBS_QUEUES_4   SUBS_QUEUES_4_SERVICES
SUBS_SESSION_SHAPER False
```

The following example displays scheduler information for a specified interface.

```
supervisor@rtbrick: op> show qos scheduler all interface ppp-0/1/23/72339069014638634
Interface Name                 Type                      Scheduler Name  Scheduler Map Name            Shaper Name
Composite
ppp-0/1/23/72339069014638634   Subscriber-Ifl            SUBS_QUEUES_4   SUBS_QUEUES_4_SERVICES
SUBS_SESSION_SHAPER False
```

The following example displays scheduler information for the specified key type.

```
supervisor@rtbrick: op> show qos scheduler all type Subscriber-Ifl
Interface Name                 Type                      Scheduler Name  Scheduler Map Name            Shaper Name
Composite
ppp-0/1/23/72339069014638637   Subscriber-Ifl            SUBS_QUEUES_4   SUBS_QUEUES_4_SERVICES
SUBS_SESSION_SHAPER False
ppp-0/1/23/72339069014638634   Subscriber-Ifl            SUBS_QUEUES_4   SUBS_QUEUES_4_SERVICES
SUBS_SESSION_SHAPER False
ppp-0/1/23/72339069014638638   Subscriber-Ifl            SUBS_QUEUES_4   SUBS_QUEUES_4_SERVICES
SUBS_SESSION_SHAPER False
ppp-0/1/23/72339069014638636   Subscriber-Ifl            SUBS_QUEUES_4   SUBS_QUEUES_4_SERVICES
SUBS_SESSION_SHAPER False
ppp-0/1/23/72339069014638635   Subscriber-Ifl            SUBS_QUEUES_4   SUBS_QUEUES_4_SERVICES
SUBS_SESSION_SHAPER False
```

#### show qos remark map

### Syntax:

**show qos remark-map** <option>

| Option | Description |
|---|---|
| `all` | Displays all remark maps configured in the system. |
| `<interface_name>` | Displays remark maps applied to a specific interface. |

| Option | Description |
|---|---|
| `type <key_type>` | Filters remark map for the specified type. |

The following example displays a summary of all the remark maps for a specified interface.

```
supervisor@rtbrick: op> show qos remark all interface ppp-0/1/23/72339069014638634
Interface: ppp-0/1/23/72339069014638634, Remark Map: SUBS_REMARKING_EGRESS, Type: Subscriber
  Remark Type          Match Codepoint     Remark Codepoint    Color
  ieee-802.1                         0                    3    all
  ieee-802.1                         2                    5    all
  ieee-802.1                        32                    3    all
  ieee-802.1                        96                    5    all
  ieee-802.1                         1                    4    all
  ieee-802.1                        64                    4    all
  ieee-802.1                       128                    6    all
  ieee-802.1                         3                    6    all
```

The following example displays a summary of all the remark maps for a specified key type.

```
supervisor@rtbrick: op> show qos remark-map all type Subscriber-Ifl
Interface: ppp-0/1/23/72339069014638638, Remark Map: SUBS_REMARKING_EGRESS, Type: Subscriber
  Remark Type          Match Codepoint     Remark Codepoint    Color
  ieee-802.1                         0                    3    all
  ieee-802.1                         2                    5    all
  ieee-802.1                        32                    3    all
  ieee-802.1                        96                    5    all
  ieee-802.1                         1                    4    all
  ieee-802.1                        64                    4    all
  ieee-802.1                       128                    6    all
  ieee-802.1                         3                    6    all
Interface: ppp-0/1/23/72339069014638634, Remark Map: SUBS_REMARKING_EGRESS, Type: Subscriber
  Remark Type          Match Codepoint     Remark Codepoint    Color
  ieee-802.1                         0                    3    all
  ieee-802.1                         2                    5    all
  ieee-802.1                        32                    3    all
  ieee-802.1                        96                    5    all
  ieee-802.1                         1                    4    all
  ieee-802.1                        64                    4    all
  ieee-802.1                       128                    6    all
  ieee-802.1                         3                    6    all
Interface: ppp-0/1/23/72339069014638635, Remark Map: SUBS_REMARKING_EGRESS, Type: Subscriber
  Remark Type          Match Codepoint     Remark Codepoint    Color
  ieee-802.1                         0                    3    all
  ieee-802.1                         2                    5    all
  ieee-802.1                        32                    3    all
  ieee-802.1                        96                    5    all
  ieee-802.1                         1                    4    all
  ieee-802.1                        64                    4    all
  ieee-802.1                       128                    6    all
  ieee-802.1                         3                    6    all
```

```
Interface: ppp-0/1/23/72339069014638637, Remark Map: SUBS_REMARKING_EGRESS, Type: Subscriber
  Remark Type          Match Codepoint     Remark Codepoint    Color
  ieee-802.1                         0                    3    all
  ieee-802.1                         2                    5    all
  ieee-802.1                        32                    3    all
  ieee-802.1                        96                    5    all
  ieee-802.1                         1                    4    all
  ieee-802.1                        64                    4    all
  ieee-802.1                       128                    6    all
  ieee-802.1                         3                    6    all
Interface: ppp-0/1/23/72339069014638636, Remark Map: SUBS_REMARKING_EGRESS, Type: Subscriber
  Remark Type          Match Codepoint     Remark Codepoint    Color
  ieee-802.1                         0                    3    all
  ieee-802.1                         2                    5    all
  ieee-802.1                        32                    3    all
  ieee-802.1                        96                    5    all
  ieee-802.1                         1                    4    all
  ieee-802.1                        64                    4    all
  ieee-802.1                       128                    6    all
  ieee-802.1                         3                    6    all
```

#### show qos scheduler-map

**Syntax:**

**show qos scheduler-map** <option>

| Option | Description |
|---|---|
| - | Without any option, this command displays a summary of all the scheduler maps. |
| <scheduler-map> | Displays scheduler information for the specified scheduler map. |

The following example displays a summary of all the scheduler maps.

```
supervisor@rtbrick: op> show qos scheduler-map
Scheduler-Map: PORT_SCHEDULER
  Scheduler: SUBS_PON_0  Type: fair_queueing
Scheduler-Map: SUBS_QUEUES_4_SERVICES
  Scheduler: SUBS_QUEUES_4  Type: strict_priority
    Queue: QUEUE_0                          Parent flow: low-flow         Connection point: strict_priority_3
    Queue: QUEUE_1                          Parent flow: high-flow        Connection point: strict_priority_1
    Queue: QUEUE_2                          Parent flow: high-flow        Connection point: strict_priority_2
    Queue: QUEUE_3                          Parent flow: high-flow        Connection point: strict_priority_0
```

The following example displays scheduler information for the specified scheduler-map.

```
supervisor@rtbrick: op> show qos scheduler SUBS_QUEUES_4
Scheduler              Type                   Shaper                 Composite              Active
```

```
SUBS_QUEUES_4          strict_priority        SUBS_SESSION_SHAPER    False                  True
```

#### show qos shaper

**Syntax:**

**show qos shaper** <option>

| Option | Description |
|---|---|
| - | Without any option, this command displays a summary of all the shapers. |
| <shaper-name> | Displays scheduler information for the specified shaper. |

The following example displays a summary of all the shapers.

```
supervisor@rtbrick: op> show qos shaper
Shaper                        High Rate(Kbps)     Precision        Low Rate(Kbps)      High Burst(Kb)      Low Burst(Kb)
Active
SUBS_PON_0_SHAPER             16000               False               2000                -                   -
False
SUBS_SESSION_SHAPER           10000               False               1000                -                   -
True
SUBS_SHAPER_QUEUE_3           1500                False               -                   -                   -
True
```

The following example displays shaper information for the specified shaper.

```
supervisor@rtbrick: op> show qos shaper SUBS_SESSION_SHAPER
Shaper                        High Rate(Kbps)     Precision        Low Rate(Kbps)      High Burst(Kb)      Low Burst(Kb)
Active
SUBS_SESSION_SHAPER           10000               False               1000                -                   -
True
```

#### show qos multifield-classifier

**Syntax:**

**show qos multifield-classifier** <option>

| Option | Description |
|---|---|
| - | Without any option, this command displays a summary of all the multifield classifiers. |
| <multifield-classifier-name> | Displays scheduler information for the specified multifield classifiers. |

The following example displays a summary of all the multifield classifiers.

```
supervisor@rtbrick>LEAF01: op> show qos multifield-classifier
Multifield Classifier: global-mfc
  ACL type: multifield_ipv4
  Ordinal: 10000
    Match:
      Source IPv4 prefix: 198.51.100.2/24
      IPv4 TOS: 224
    Action:
      Forward class: class-7
  Priority: 1000
  Ordinal: 9000
    Match:
      Source IPv4 prefix: 198.51.100.2/24
      IPv4 TOS: 192
    Action:
      Remark codepoint: 184
      Forward class: class-6
  Ordinal: 6200
    Match:
      Source IPv4 prefix: 198.51.100.2/24
      IPv4 TOS: 160
    Action:
      Remark codepoint: 184
      Forward class: class-5
  Ordinal: 5200
    Match:
      Source IPv4 prefix: 198.51.100.2/24
      IPv4 TOS: 64
    Action:
      Remark codepoint: 184
      Forward class: class-2
  Ordinal: 6000
    Match:
      Source IPv4 prefix: 198.51.100.2/24
      IPv4 TOS: 96
    Action:
      Remark codepoint: 184
      Forward class: class-3
  Ordinal: 5000
    Match:
      Source IPv4 prefix: 198.51.100.2/24
    Action:
      Remark codepoint: 184
      Forward class: class-0
  Ordinal: 6100
    Match:
      Source IPv4 prefix: 198.51.100.2/24
      IPv4 TOS: 128
    Action:
      Remark codepoint: 184
```

```
      Forward class: class-4
  Ordinal: 5100
    Match:
      Source IPv4 prefix: 198.51.100.2/24
      IPv4 TOS: 32
    Action:
      Remark codepoint: 184
      Forward class: class-1
  Ordinal: 100
    Match:
      Source IPv6 prefix: 2001:db8:0:100::/32
      IPv6 TC: 224
    Action:
      Remark codepoint: 196
      Forward class: class-3
Multifield Classifier: ipoe-double-play
  ACL type: multifield_ipv4
  Ordinal: 5200
    Match:
      Destination IPv4 prefix: 198.51.100.2/24
      IPv4 TOS: 160
    Action:
      Remark codepoint: 224
      Forward class: class-2
  Ordinal: 5000
    Match:
      Destination IPv4 prefix: 198.51.100.2/24
      IPv4 TOS: 64
    Action:
      Remark codepoint: 224
      Forward class: class-0
  Ordinal: 5300
    Match:
      Destination IPv4 prefix: 198.51.100.2/24
      IPv4 TOS: 192
    Action:
      Remark codepoint: 224
      Forward class: class-3
  Ordinal: 5100
    Match:
      Destination IPv4 prefix: 198.51.100.2/24
      IPv4 TOS: 96
    Action:
      Remark codepoint: 224
      Forward class: class-1
supervisor@rtbrick>LEAF01: op>
```

The following example displays multifield-classifier information for the specified multifield-classifier.

```
supervisor@rtbrick>LEAF01: op> show qos multifield-classifier global-mfc
Multifield Classifier: global-mfc
  ACL type: multifield_ipv4
  Ordinal: 10000
    Match:
      Source IPv4 prefix: 198.51.100.2/24
      IPv4 TOS: 224
    Action:
      Forward class: class-7
  Priority: 1000
  Ordinal: 9000
    Match:
      Source IPv4 prefix: 198.51.100.2/24
      IPv4 TOS: 192
    Action:
      Remark codepoint: 184
      Forward class: class-6
  Ordinal: 6200
    Match:
      Source IPv4 prefix: 198.51.100.2/24
      IPv4 TOS: 160
    Action:
      Remark codepoint: 184
      Forward class: class-5
  Ordinal: 5200
    Match:
      Source IPv4 prefix: 198.51.100.2/24
      IPv4 TOS: 64
    Action:
      Remark codepoint: 184
      Forward class: class-2
supervisor@rtbrick>LEAF01: op>
```

**Displaying QoS Statistics for Layer-2 and Layer-3 Logical Interfaces**

> Due to hardware limitations in the ASIC, the QoS dropped counters may show inaccurate values during times of congestion. At present, only the forward counters can be considered reliable.

To display QoS Statistics for layer-2 and layer-3 logical interfaces, enter the following command:

**show interface** <layer-2/layer-3 interface> **statistics**

To obtain detailed view of QoS statistics for layer-2 and layer-3 logical interfaces, enter the following command:

**show interface** <layer-2/layer-3 interface> **detail**

To clear QoS statistics of layer-2 and layer-3 logical interfaces, enter the following command:

**clear interface statistics** <layer-2/layer-3 interface>

The following example displays QoS Statistics for the interface `ifl-0/1/24/200`.

```
supervisor@rtbrick: op> show interface ifl-0/1/24/200 statistics
Logical Interface:  ifl-0/1/24/200
   VPP statistics:
     Counter              Unit      Count
     Rx                 packets   0
                        bytes     0
     Tx                 packets   14
                        bytes     1300
     IPv4               packets   0
     IPv6               packets   0
     MPLS               packets   0
     Punt               packets   0
     Drops              packets   0
     Rx Miss            packets   0
     Rx Error           packets   0
     Rx No Buff         packets   0
     Tx Error           packets   0
  BCM statistics:
    Ingress forwarded packets: 248              Ingress forwarded bytes: 248992
    Ingress drop packets: 518                   Ingress drop bytes: 520072
    Egress forwarded packets: 0                 Egress forwarded bytes: 0
    Egress drop packets: 28404                  Egress drop bytes: 28915272
    Egress Class (Queue) Statistics:
        class-0: 0 packets 0 bytes dropped: 12668 packets 12896024 bytes
        class-1: 0 packets 0 bytes dropped: 0 packets 0 bytes
        class-2: 0 packets 0 bytes dropped: 4664 packets 4747952 bytes
        class-3: 0 packets 0 bytes dropped: 11072 packets 11271296 bytes
        class-4: 0 packets 0 bytes dropped: 0 packets 0 bytes
        class-5: 0 packets 0 bytes dropped: 0 packets 0 bytes
        class-6: 0 packets 0 bytes dropped: 0 packets 0 bytes
        class-7: 0 packets 0 bytes dropped: 0 packets 0 bytes
    Ingress Policer Statistics:
        Level 1: 14 packets 14056 bytes dropped: 69 packets 69276 bytes
        Level 2: 81 packets 81324 bytes dropped: 165 packets 165660 bytes
        Level 3: 144 packets 144576 bytes dropped: 191 packets 191764 bytes
        Level 4: 9 packets 9036 bytes dropped: 93 packets 93372 bytes
```

The following example displays detailed QoS Statistics for the interface `ifl-0/1/24/200`.

```
supervisor@rtbrick: op> show interface ifl-0/1/24/200 detail
  Interface: ifl-0/1/24/200, Instance: default
    Admin/Link/Operational status: Up/Up/Up
    IPv4/IPv6/MPLS Status: Up/Up/Up
    IPv4/IPv6/MPLS MTU: 1500/1500/1500
    Interface type: Logical Sub interface
```

```
    Vlan Encapsulation: 802.1q
    Outer VLAN: 200
    RPF Mode: disable
    Neighbor Preference: 6
    Interface index: 52453505
    MAC: 5c:07:58:a2:cb:96
    Class-of-Service: SUBSCRIBER_QOS_PROFILE
    IPv4 Address             IPv6 Address
    192.0.2.9/30             fe80::5c07:5800:c8a2:cb96/64
    BCM statistics:
      Ingress forwarded packets: 248                Ingress forwarded bytes: 248992
      Ingress drop packets: 518                     Ingress drop bytes: 520072
      Egress forwarded packets: 0                   Egress forwarded bytes: 0
      Egress drop packets: 28404                    Egress drop bytes: 28915272
    Egress Class (Queue) Statistics:
        class-0: 0 packets 0 bytes dropped: 12668 packets 12896024 bytes
        class-1: 0 packets 0 bytes dropped: 0 packets 0 bytes
        class-2: 0 packets 0 bytes dropped: 4664 packets 4747952 bytes
        class-3: 0 packets 0 bytes dropped: 11072 packets 11271296 bytes
        class-4: 0 packets 0 bytes dropped: 0 packets 0 bytes
        class-5: 0 packets 0 bytes dropped: 0 packets 0 bytes
        class-6: 0 packets 0 bytes dropped: 0 packets 0 bytes
        class-7: 0 packets 0 bytes dropped: 0 packets 0 bytes
    Ingress Policer Statistics:
        Level 1: 14 packets 14056 bytes dropped: 69 packets 69276 bytes
        Level 2: 81 packets 81324 bytes dropped: 165 packets 165660 bytes
        Level 3: 144 packets 144576 bytes dropped: 191 packets 191764 bytes
        Level 4: 9 packets 9036 bytes dropped: 93 packets 93372 bytes
```

# 6.8. LLDP

## 6.8.1. LLDP Overview

Link Layer Discovery Protocol (LLDP) is a media-independent link layer protocol used by network devices for advertising their identity, capabilities to neighbors on a LAN segment. LLDP runs over the data-link layer only, allowing two systems running different network layer protocols to learn about each other.

LLDP supports a set of attributes that it uses to discover neighbor devices. These attributes contain type, length, and value descriptions and are referred to as TLVs. LLDP supported devices can use TLVs to receive and send information to their neighbors. Details such as configuration information, device capabilities, and device identity can be advertised using this protocol.

**Guidelines**

- All LLDP packets are sent to the CPU for further processing.
- LLDP packets will be lower priority packets when compared with the routing protocol packets. In case of a congestion, the LLDP packets may be policed.

### Limitations

- LLDP currently does not intend to work on bundle interfaces.
- LLDP is disabled for the following interfaces on L2BSA L2X:
  - Physical interfaces on Q2C platform
  - LAG member interfaces on QAX platform

### Interactions with other features

RBFS provides various daemons that run as background processes. LLDP interacts with these daemons for accessing necessary configurations.

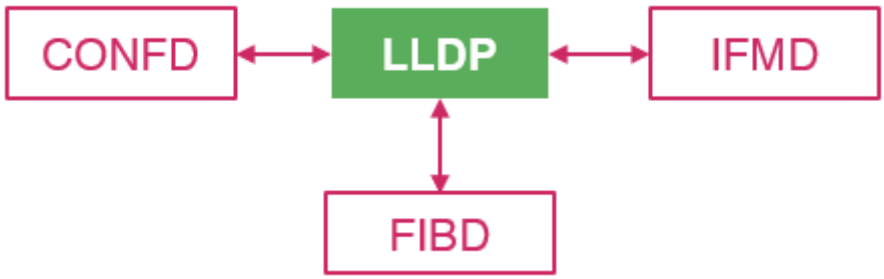



*Figure 145. LLDP Interactions*

The table below shows the interactions of LLDP with the other daemons of RBFS.

*Interactions of LLDP with other daemons*

| Daemon Name | Daemon Description | Interaction Details |
|---|---|---|
| Confd | Configuration | *Confd* stores all the LLDP configuration information. LLDP subscribes for the following tables of *Confd*:<br>• global.lldp.config<br>• global.lldp.interface.config |
| IFMD | Interface Monitoring | LLDP is enabled on the physical interface. LLDP subscribes to the IFP table to find the status of the physical link. An LLDP message can be sent only to those interfaces whose link state is UP. |
| FIBD | Forwarding Information Base | *FIBD* subscribes for the LLDP interface table which has all the information required for enabling LLDP on the interface. This is used to program in VPP. Any neighbor detected, VPP will notify and this detection is stored in the FIBD table for tracking purpose. Each time a message is received from the neighbor, it will be updated in this table. Thus, the history of the messages received from the neighbor is tracked. This information is stored in one of the FIBD tables. LLDP subscribes to this table to find its neighbor. |

#### Default Inclusion of System Description TLV in LLDP

The System Description TLV is included in LLDP packets and contains details about the RBFS software, its version, and hardware platform.

#### LLDP Limitations with L2X

LLDP is disabled in the following scenario:

1. The outgoing-interface is configured in the egress L2X.
2. The outgoing-interface is configured in the ingress L2X with the match-type as "any" or "untagged".
3. The outgoing-interface and incoming-interface are configured in the bi-directional L2X.

### Supported Platforms

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

## 6.8.2. LLDP Configuration

### Configuration Hierarchy

The diagram illustrates the LLDP configuration hierarchy.

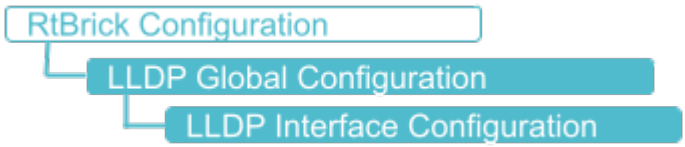



*Figure 146. LLDP Configuration Hierarchy*

### Configuration Syntax and Commands

The following sections describe the LLDP configuration syntax and commands.

#### LLDP Global Configuration

**Syntax:**

**set lldp** <attribute> <value>

| Attribute | Description |
|---|---|
| `admin-status [disable\|enable]` | Enable or disable LLDP. LLDP is enabled, by default. |
| `system-description <system description>` | LLDP global system description to be sent to the neighbor |
| `system-name <system name>` | LLDP global system name to be sent to the neighbor. If the system name is not configured, it is fetched from BDS. |

| Attribute | Description |
|---|---|
| `tx-hold <transmit hold time>` | Specifies the amount of time (in seconds) a receiving device maintains the neighbor information before aging your device. If the timer expires and no LLPD packet was received, the neighbor will be marked as DOWN. Default value is 120 seconds. The hold-time range is 1 through 360000. |
| `tx-interval <advertisement interval>` | Interval (in seconds) at which LLDP packets are sent to neighbors. Default interval value is 30 seconds. The transmission interval range is 1 through 3600 seconds. |

Example 1: Enable LLDP

```
{
    "rtbrick-config:lldp": {
      "admin-status": "enable",
      "system-name": "rtbrick",
      "system-description": "This is rtbrick system",
      "tx-interval": 40,
      "tx-hold": 150
    }
  }
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### LLDP Interface Configuration

**Syntax:**

**set lldp interface** <interface-name> <attribute> <value>

| Attribute | Description |
|---|---|
| `interface <interface name>` | Name of the interface on which you enable or disable LLDP. By default, the interface is enabled.<br>NOTE: Interface level configuration will override the global LLDP enable/disable functionality. |
| `admin-status [disable or enable]` | LLDP is enabled by default. The command *set lldp admin-status disable* is used to disable it. If you want to re-enable it, run the *set lldp admin-status enable* command. |
| `port-desc <port description>` | LLDP port description to be sent to the neighbor. If the port description is not configured, the description configured under interface from IFMD is taken as LLDP port description. |

Example 1: LLDP on Interface Configuration

```
{
    "rtbrick-config:lldp": {
      "interface": [
        {
          "interface-name": "ifp-0/0/1",
          "port-description": "this is port ifp-0/0/1",
          "admin-status": "enable"
        }
      ]
    }
  }
```

API To access the RESTCONF API that corresponds to this CLI, click here.

### 6.8.3. LLDP Operational Commands

#### LLDP Show Commands

The LLDP show commands provide detailed information about the LLDP global summary, neighbors, and interfaces.

**Syntax:**

**show lldp** <option>

| Option | Description |
|---|---|
| `interface` | Displays information about interfaces where LLDP is enabled. |
| `neighbor` | Displays information about all the neighbors. |
| `summary` | Displays global summary details such as system name, frequency of transmissions, the hold-time for the packets sent, TLVs, and the disabled TLVs. |

##### LLDP Interfaces

The command displays the information about interfaces where LLDP is enabled.

**Syntax:**

**show lldp interface** <attribute> <value>

| Option | Description |
|---|---|
| `interface-name` | Displays summary details such as status, MAC address and description. |

Example 1: LLDP interface summary

```
supervisor@rtbrick: op> show lldp interface
Interface name      Status    MAC address        Description
ifp-0/1/0           Up        5c:07:58:79:0a:00  Physical interface #0 from node 0, chip 1
ifp-0/1/1           Down      5c:07:58:79:0a:00  Physical interface #1 from node 0, chip 1
ifp-0/1/2           Up        5c:07:58:79:0a:00  Physical interface #2 from node 0, chip 1
ifp-0/1/3           Up        5c:07:58:79:0a:00  Physical interface #3 from node 0, chip 1
ifp-0/1/4           Down      5c:07:58:79:0a:00  Physical interface #4 from node 0, chip 1
ifp-0/1/5           Up        5c:07:58:79:0a:00  Physical interface #5 from node 0, chip 1
ifp-0/1/6           Up        5c:07:58:79:0a:00  Physical interface #6 from node 0, chip 1
ifp-0/1/7           Up        5c:07:58:79:0a:00  Physical interface #7 from node 0, chip 1
ifp-0/1/8           Up        5c:07:58:79:0a:00  Physical interface #8 from node 0, chip 1
ifp-0/1/9           Down      5c:07:58:79:0a:00  Physical interface #9 from node 0, chip 1
ifp-0/1/10          Up        5c:07:58:79:0a:00  Physical interface #10 from node 0, chip 1
ifp-0/1/11          Down      5c:07:58:79:0a:00  Physical interface #11 from node 0, chip 1
ifp-0/1/12          Up        5c:07:58:79:0a:00  Physical interface #12 from node 0, chip 1
ifp-0/1/13          Down      5c:07:58:79:0a:00  Physical interface #13 from node 0, chip 1
ifp-0/1/14          Up        5c:07:58:79:0a:00  Physical interface #14 from node 0, chip 1
ifp-0/1/15          Down      5c:07:58:79:0a:00  Physical interface #15 from node 0, chip 1
ifp-0/1/16          Up        5c:07:58:79:0a:00  Physical interface #16 from node 0, chip 1
ifp-0/1/17          Down      5c:07:58:79:0a:00  Physical interface #17 from node 0, chip 1
ifp-0/1/18          Down      5c:07:58:79:0a:00  Physical interface #18 from node 0, chip 1
ifp-0/1/19          Down      5c:07:58:79:0a:00  Physical interface #19 from node 0, chip 1
ifp-0/1/20          Up        5c:07:58:79:0a:00  Physical interface #20 from node 0, chip 1
ifp-0/1/21          Up        5c:07:58:79:0a:00  Physical interface #21 from node 0, chip 1
ifp-0/1/22          Up        5c:07:58:79:0a:00  Physical interface #22 from node 0, chip 1
ifp-0/1/23          Down      5c:07:58:79:0a:00  Physical interface #23 from node 0, chip 1
ifp-0/1/24          Up        5c:07:58:79:0a:00  Physical interface #24 from node 0, chip 1
ifp-0/1/25          Up        5c:07:58:79:0a:00  Physical interface #25 from node 0, chip 1
ifp-0/1/26          Up        5c:07:58:79:0a:00  Physical interface #26 from node 0, chip 1
ifp-0/1/27          Down      5c:07:58:79:0a:00  Physical interface #27 from node 0, chip 1
ifp-0/1/28          Down      5c:07:58:79:0a:00  Physical interface #28 from node 0, chip 1
ifp-0/1/29          Down      5c:07:58:79:0a:00  Physical interface #29 from node 0, chip 1
ifp-0/1/30          Down      5c:07:58:79:0a:00  Physical interface #30 from node 0, chip 1
ifp-0/1/31          Down      5c:07:58:79:0a:00  Physical interface #31 from node 0, chip 1
ifp-0/1/32          Up        5c:07:58:79:0a:00  Physical interface #32 from node 0, chip 1
ifp-0/1/33          Up        5c:07:58:79:0a:00  Physical interface #33 from node 0, chip 1
ifp-0/1/34          Down      5c:07:58:79:0a:00  Physical interface #34 from node 0, chip 1
ifp-0/1/35          Down      5c:07:58:79:0a:00  Physical interface #35 from node 0, chip 1
ifp-0/1/36          Down      5c:07:58:79:0a:00  Physical interface #36 from node 0, chip 1
ifp-0/1/37          Down      5c:07:58:79:0a:00  Physical interface #37 from node 0, chip 1
ifp-0/1/38          Down      5c:07:58:79:0a:00  Physical interface #38 from node 0, chip 1
ifp-0/1/39          Down      5c:07:58:79:0a:00  Physical interface #39 from node 0, chip 1
ifp-0/1/40          Down      5c:07:58:79:0a:00  Physical interface #40 from node 0, chip 1
ifp-0/1/41          Up        5c:07:58:79:0a:00  Physical interface #41 from node 0, chip 1
ifp-0/1/42          Up        5c:07:58:79:0a:00  Physical interface #42 from node 0, chip 1
ifp-0/1/43          Down      5c:07:58:79:0a:00  Physical interface #43 from node 0, chip 1
ifp-0/1/44          Down      5c:07:58:79:0a:00  Physical interface #44 from node 0, chip 1
ifp-0/1/45          Down      5c:07:58:79:0a:00  Physical interface #45 from node 0, chip 1
ifp-0/1/46          Up        5c:07:58:79:0a:00  Physical interface #46 from node 0, chip 1
ifp-0/1/47          Up        5c:07:58:79:0a:00  Physical interface #47 from node 0, chip 1
ifp-0/1/48          Up        5c:07:58:79:0a:00  Physical interface #48 from node 0, chip 1
```

```
ifp-0/1/49          Up        5c:07:58:79:0a:00  Physical interface #49 from node 0, chip 1
ifp-0/1/50          Down      5c:07:58:79:0a:00  Physical interface #50 from node 0, chip 1
ifp-0/1/51          Down      5c:07:58:79:0a:00  Physical interface #51 from node 0, chip 1
ifp-0/1/52          Down      5c:07:58:79:0a:00  Physical interface #52 from node 0, chip 1
ifp-0/1/53          Down      5c:07:58:79:0a:00  Physical interface #53 from node 0, chip 1
ifp-0/1/54          Down      5c:07:58:79:0a:00  Physical interface #54 from node 0, chip 1
ifp-0/1/55          Down      5c:07:58:79:0a:00  Physical interface #55 from node 0, chip 1
ifp-0/1/56          Down      5c:07:58:79:0a:00  Physical interface #56 from node 0, chip 1
ifp-0/1/57          Down      5c:07:58:79:0a:00  Physical interface #57 from node 0, chip 1
ifp-0/1/58          Down      5c:07:58:79:0a:00  Physical interface #58 from node 0, chip 1
ifp-0/1/59          Down      5c:07:58:79:0a:00  Physical interface #59 from node 0, chip 1
ifp-0/1/60          Down      5c:07:58:79:0a:00  Physical interface #60 from node 0, chip 1
ifp-0/1/61          Down      5c:07:58:79:0a:00  Physical interface #61 from node 0, chip 1
ifp-0/1/62          Down      5c:07:58:79:0a:00  Physical interface #62 from node 0, chip 1
ifp-0/1/63          Down      5c:07:58:79:0a:00  Physical interface #63 from node 0, chip 1
ifp-0/1/64          Down      5c:07:58:79:0a:00  Physical interface #64 from node 0, chip 1
ifp-0/1/65          Down      5c:07:58:79:0a:00  Physical interface #65 from node 0, chip 1
ifp-0/1/66          Down      5c:07:58:79:0a:00  Physical interface #66 from node 0, chip 1
ifp-0/1/67          Down      5c:07:58:79:0a:00  Physical interface #67 from node 0, chip 1
ifp-0/1/68          Down      5c:07:58:79:0a:00  Physical interface #68 from node 0, chip 1
ifp-0/1/69          Down      5c:07:58:79:0a:00  Physical interface #69 from node 0, chip 1
ifp-0/1/70          Down      5c:07:58:79:0a:00  Physical interface #70 from node 0, chip 1
ifp-0/1/71          Down      5c:07:58:79:0a:00  Physical interface #71 from node 0, chip 1
ifp-0/1/72          Down      5c:07:58:79:0a:00  Physical interface #72 from node 0, chip 1
ifp-0/1/73          Down      5c:07:58:79:0a:00  Physical interface #73 from node 0, chip 1
ifp-0/1/74          Down      5c:07:58:79:0a:00  Physical interface #74 from node 0, chip 1
ifp-0/1/75          Down      5c:07:58:79:0a:00  Physical interface #75 from node 0, chip 1
ifp-0/1/76          Down      5c:07:58:79:0a:00  Physical interface #76 from node 0, chip 1
ifp-0/1/77          Down      5c:07:58:79:0a:00  Physical interface #77 from node 0, chip 1
ifp-0/1/78          Down      5c:07:58:79:0a:00  Physical interface #78 from node 0, chip 1
ifp-0/1/79          Down      5c:07:58:79:0a:00  Physical interface #79 from node 0, chip 1
ifp-0/1/80          Down      5c:07:58:79:0a:00  Physical interface #80 from node 0, chip 1
ifp-0/1/81          Down      5c:07:58:79:0a:00  Physical interface #81 from node 0, chip 1
ifp-0/1/82          Down      5c:07:58:79:0a:00  Physical interface #82 from node 0, chip 1
ifp-0/1/83          Down      5c:07:58:79:0a:00  Physical interface #83 from node 0, chip 1
ifp-0/1/84          Down      5c:07:58:79:0a:00  Physical interface #84 from node 0, chip 1
ifp-0/1/85          Down      5c:07:58:79:0a:00  Physical interface #85 from node 0, chip 1
ifp-0/1/86          Down      5c:07:58:79:0a:00  Physical interface #86 from node 0, chip 1
ifp-0/1/87          Down      5c:07:58:79:0a:00  Physical interface #87 from node 0, chip 1
ifp-0/1/88          Down      5c:07:58:79:0a:00  Physical interface #88 from node 0, chip 1
ifp-0/1/89          Down      5c:07:58:79:0a:00  Physical interface #89 from node 0, chip 1
ifp-0/1/90          Down      5c:07:58:79:0a:00  Physical interface #90 from node 0, chip 1
ifp-0/1/91          Down      5c:07:58:79:0a:00  Physical interface #91 from node 0, chip 1
ifp-0/1/92          Down      5c:07:58:79:0a:00  Physical interface #92 from node 0, chip 1
ifp-0/1/93          Down      5c:07:58:79:0a:00  Physical interface #93 from node 0, chip 1
ifp-0/1/94          Down      5c:07:58:79:0a:00  Physical interface #94 from node 0, chip 1
ifp-0/1/95          Down      5c:07:58:79:0a:00  Physical interface #95 from node 0, chip 1
ifp-0/1/96          Up        5c:07:58:79:0a:00  Physical interface #96 from node 0, chip 1
ifp-0/1/97          Up        5c:07:58:79:0a:00  Physical interface #97 from node 0, chip 1
ifp-0/1/98          Down      5c:07:58:79:0a:00  Physical interface #98 from node 0, chip 1
ifp-0/1/99          Down      5c:07:58:79:0a:00  Physical interface #99 from node 0, chip 1
ifp-0/1/100         Down      5c:07:58:79:0a:00  Physical interface #100 from node 0, chip 1
ifp-0/1/101         Up        5c:07:58:79:0a:00  Physical interface #101 from node 0, chip 1
```

#### LLDP Neighbors

This command displays the information of all neighbors.

**Syntax:**

**show lldp neighbor** <attribute> <value>

| Option | Description |
|---|---|
| `detail` | Displays the information in detail about a specific LLDP neighbor or all neighbors. |
| `<interface-name>` | Name of the interface on which neighbor is formed. |

Example 1: LLDP neighbor summary

```
supervisor@rtbrick: op> show lldp neighbor
Neighbor name         Status  Remote port ID        Local port ID      Neighbor MAC address  Last received     Last sent
ufi18.q2c.u36.r6....  Up      ifp-0/1/0             ifp-0/1/0          5c:07:58:79:0f:00     0:00:15 ago       0:00:19
ago
ufi18.q2c.u36.r6....  Up      ifp-0/1/16            ifp-0/1/16         5c:07:58:79:0f:00     0:00:15 ago       0:00:19
ago
ds60.r6.nbg           Up      568                   ifp-0/1/2          84:b5:9c:49:c9:c0     0:00:08 ago       0:00:19
ago
ufi10.q2c.u9.r4.n...  Up      ifp-0/1/20            ifp-0/1/22         e8:c5:7a:8f:56:0e     0:00:12 ago       0:00:19
ago
n/a                   Up      00:00:00:00:01:01     ifp-0/1/24         00:00:00:00:01:01     0:00:16 ago       0:00:19
ago
n/a                   Up      00:00:00:00:01:00     ifp-0/1/25         00:00:00:00:01:00     0:00:16 ago       0:00:19
ago
n/a                   Up      3c:fd:fe:b4:86:fb     ifp-0/1/26         3c:fd:fe:b4:86:fb     0:00:16 ago       0:00:19
ago
ds60.r6.nbg           Up      566                   ifp-0/1/3          84:b5:9c:49:c9:c0     0:00:13 ago       0:00:19
ago
mx2.r6.nbg.rtrick.net  Up      xe-0/0/0              ifp-0/1/32         3c:8a:b0:8a:4c:28     0:00:25 ago       0:00:19
ago
mx2.r6.nbg.rtrick.net  Up      xe-0/0/1              ifp-0/1/33         3c:8a:b0:8a:4c:28     0:00:22 ago       0:00:19
ago
exp61.u26.r6.nbg....  Up      ifp-0/0/41            ifp-0/1/41         b8:6a:97:e6:2e:1b     0:00:21 ago       0:00:19
ago
exp61.u26.r6.nbg....  Up      ifp-0/0/42            ifp-0/1/42         b8:6a:97:e6:2e:1b     0:00:21 ago       0:00:19
ago
ufi21.q2a.u17.r6....  Up      ifp-0/0/26            ifp-0/1/46         5c:07:58:78:90:90     0:00:20 ago       0:00:19
ago
ufi21.q2a.u17.r6....  Up      ifp-0/0/27            ifp-0/1/47         5c:07:58:78:90:90     0:00:20 ago       0:00:19
ago
n/a                   Up      3c:fd:fe:d0:22:82     ifp-0/1/48         3c:fd:fe:d0:22:82     0:00:18 ago       0:00:19
ago
n/a                   Up      3c:fd:fe:d0:22:80     ifp-0/1/49         3c:fd:fe:d0:22:80     0:00:18 ago       0:00:19
ago
ufi19.q2a.u19.r6....  Up      ifp-0/0/5             ifp-0/1/5          5c:07:58:78:85:48     0:00:20 ago       0:00:19
ago
ufi19.q2a.u19.r6....  Up      ifp-0/0/6             ifp-0/1/6          5c:07:58:78:85:48     0:00:20 ago       0:00:19
ago
ufi20.q2a.u18.r6....  Up      ifp-0/0/7             ifp-0/1/7          5c:07:58:78:84:fc     0:00:20 ago       0:00:19
ago
ufi20.q2a.u18.r6....  Up      ifp-0/0/8             ifp-0/1/8          5c:07:58:78:84:fc     0:00:20 ago       0:00:19
ago
ufi18.q2c.u36.r6....  Up      ifp-0/1/96            ifp-0/1/96         5c:07:58:79:0f:00     0:00:20 ago       0:00:19
```

```
ago
ufi18.q2c.u36.r6....  Up      ifp-0/1/97            ifp-0/1/97         5c:07:58:79:0f:00     0:00:20 ago       0:00:19
ago
```

### Example 2: LLDP all neighbor details

```
supervisor@rtbrick: op> show lldp neighbor detail
Neighbor: ufi18.q2c.u36.r6.nbg.rtbrick.net
  Neighbor MAC address: 5c:07:58:79:0f:00
  Neighbor port ID: ifp-0/1/0
  Neighbor port description: Physical interface #0 from node 0, chip 1
  Neighbor system description: Rtbrick, Inc. s9600-102xc router, version 26.2.0-
g6daily.20260604183222+Bdevelopment.C76bb84e8, Build date: 2026-06-04 18:32:22 UTC
  Neighbor TTL: 121
  Local interface: ifp-0/1/0
  Local MAC address: 5c:07:58:79:0a:00
  Local port description: Physical interface #0 from node 0, chip 1
  Packets sent: 1
  Packets received: 13
  Neighbor status: Up
Neighbor: ufi18.q2c.u36.r6.nbg.rtbrick.net
  Neighbor MAC address: 5c:07:58:79:0f:00
  Neighbor port ID: ifp-0/1/16
  Neighbor port description: Physical interface #16 from node 0, chip 1
  Neighbor system description: Rtbrick, Inc. s9600-102xc router, version 26.2.0-
g6daily.20260604183222+Bdevelopment.C76bb84e8, Build date: 2026-06-04 18:32:22 UTC
  Neighbor TTL: 121
  Local interface: ifp-0/1/16
  Local MAC address: 5c:07:58:79:0a:00
  Local port description: Physical interface #16 from node 0, chip 1
  Packets sent: 1
  Packets received: 13
  Neighbor status: Up
Neighbor: ds60.r6.nbg
  Neighbor MAC address: 84:b5:9c:49:c9:c0
  Neighbor port ID: 568
  Neighbor port description: ufi17 - ifp-0/1/2
  Neighbor system description: Juniper Networks, Inc. ex4550-32f Ethernet Switch, kernel JUNOS 15.1R6.7, Build date:
2017-04-23 01:16:48 UTC Copyright (c) 1996-2017 Juniper Networks, Inc.
  Neighbor TTL: 120
  Local interface: ifp-0/1/2
  Local MAC address: 5c:07:58:79:0a:00
  Local port description: Physical interface #2 from node 0, chip 1
  Packets sent: 1
  Packets received: 19
  Neighbor status: Up
Neighbor: ufi10.q2c.u9.r4.nbg.rtbrick.net
  Neighbor MAC address: e8:c5:7a:8f:56:0e
  Neighbor port ID: ifp-0/1/20
  Neighbor port description: Physical interface #20 from node 0, chip 1
  Neighbor system description: Rtbrick, Inc. s9600-32x router, version 26.2.1-candidate.4, Build date: 2026-06-03
14:28:28 UTC
  Neighbor TTL: 121
  Local interface: ifp-0/1/22
  Local MAC address: 5c:07:58:79:0a:00
  Local port description: Physical interface #22 from node 0, chip 1
  Packets sent: 1
  Packets received: 1
  Neighbor status: Up
Neighbor:
  Neighbor MAC address: 00:00:00:00:01:01
  Neighbor port ID: 00:00:00:00:01:01
  Neighbor TTL: 121
  Local interface: ifp-0/1/24
```

```
  Local MAC address: 5c:07:58:79:0a:00
  Local port description: Physical interface #24 from node 0, chip 1
  Packets sent: 1
  Packets received: 5
  Neighbor status: Up
Neighbor:
  Neighbor MAC address: 00:00:00:00:01:00
  Neighbor port ID: 00:00:00:00:01:00
  Neighbor TTL: 121
  Local interface: ifp-0/1/25
  Local MAC address: 5c:07:58:79:0a:00
  Local port description: Physical interface #25 from node 0, chip 1
  Packets sent: 1
  Packets received: 5
  Neighbor status: Up
Neighbor:
  Neighbor MAC address: 3c:fd:fe:b4:86:fb
  Neighbor port ID: 3c:fd:fe:b4:86:fb
  Neighbor TTL: 121
  Local interface: ifp-0/1/26
  Local MAC address: 5c:07:58:79:0a:00
  Local port description: Physical interface #26 from node 0, chip 1
  Packets sent: 1
  Packets received: 4
  Neighbor status: Up
Neighbor: ds60.r6.nbg
  Neighbor MAC address: 84:b5:9c:49:c9:c0
  Neighbor port ID: 566
  Neighbor port description: ufi17 - ifp-0/1/3
  Neighbor system description: Juniper Networks, Inc. ex4550-32f Ethernet Switch, kernel JUNOS 15.1R6.7, Build date:
2017-04-23 01:16:48 UTC Copyright (c) 1996-2017 Juniper Networks, Inc.
  Neighbor TTL: 120
  Local interface: ifp-0/1/3
  Local MAC address: 5c:07:58:79:0a:00
  Local port description: Physical interface #3 from node 0, chip 1
  Packets sent: 1
  Packets received: 19
  Neighbor status: Up
Neighbor: mx2.r6.nbg.rtrick.net
  Neighbor MAC address: 3c:8a:b0:8a:4c:28
  Neighbor port ID: xe-0/0/0
  Neighbor port description: xe-0/0/0
  Neighbor system description: Juniper Networks, Inc. mx5-t internet router, kernel JUNOS 18.4R2-S3, Build date: 2019-12-
26 21:41:02 UTC Copyright (c) 1996-2019 Juniper Networks, Inc.
  Neighbor TTL: 120
  Local interface: ifp-0/1/32
  Local MAC address: 5c:07:58:79:0a:00
  Local port description: Physical interface #32 from node 0, chip 1
  Packets sent: 1
  Packets received: 14
  Neighbor status: Up
Neighbor: mx2.r6.nbg.rtrick.net
  Neighbor MAC address: 3c:8a:b0:8a:4c:28
  Neighbor port ID: xe-0/0/1
  Neighbor port description: xe-0/0/1
  Neighbor system description: Juniper Networks, Inc. mx5-t internet router, kernel JUNOS 18.4R2-S3, Build date: 2019-12-
26 21:41:02 UTC Copyright (c) 1996-2019 Juniper Networks, Inc.
  Neighbor TTL: 120
  Local interface: ifp-0/1/33
  Local MAC address: 5c:07:58:79:0a:00
  Local port description: Physical interface #33 from node 0, chip 1
  Packets sent: 1
  Packets received: 14
  Neighbor status: Up
Neighbor: exp61.u26.r6.nbg.rtbrick.net
  Neighbor MAC address: b8:6a:97:e6:2e:1b
```

```
    Neighbor port ID: ifp-0/0/41
    Neighbor port description: Physical interface #41 from node 0, chip 0
    Neighbor system description: exp61.u26.r6.nbg.rtbrick.net
    Neighbor TTL: 121
    Local interface: ifp-0/1/41
    Local MAC address: 5c:07:58:79:0a:00
    Local port description: Physical interface #41 from node 0, chip 1
    Packets sent: 1
    Packets received: 12
    Neighbor status: Up
  Neighbor: exp61.u26.r6.nbg.rtbrick.net
    Neighbor MAC address: b8:6a:97:e6:2e:1b
    Neighbor port ID: ifp-0/0/42
    Neighbor port description: Physical interface #42 from node 0, chip 0
    Neighbor system description: exp61.u26.r6.nbg.rtbrick.net
    Neighbor TTL: 121
    Local interface: ifp-0/1/42
    Local MAC address: 5c:07:58:79:0a:00
    Local port description: Physical interface #42 from node 0, chip 1
    Packets sent: 1
    Packets received: 12
    Neighbor status: Up
  Neighbor: ufi21.q2a.u17.r6.nbg.rtbrick.net
    Neighbor MAC address: 5c:07:58:78:90:90
    Neighbor port ID: ifp-0/0/26
    Neighbor port description: Physical interface #26 from node 0, chip 0
    Neighbor system description: Rtbrick, Inc. s9510-28dc router, version 26.2.0-
  g6daily.20260604183218+Bdevelopment.C76bb84e8, Build date: 2026-06-04 18:32:18 UTC
    Neighbor TTL: 121
    Local interface: ifp-0/1/46
    Local MAC address: 5c:07:58:79:0a:00
    Local port description: Physical interface #46 from node 0, chip 1
    Packets sent: 1
    Packets received: 12
    Neighbor status: Up
  Neighbor: ufi21.q2a.u17.r6.nbg.rtbrick.net
    Neighbor MAC address: 5c:07:58:78:90:90
    Neighbor port ID: ifp-0/0/27
    Neighbor port description: Physical interface #27 from node 0, chip 0
    Neighbor system description: Rtbrick, Inc. s9510-28dc router, version 26.2.0-
  g6daily.20260604183218+Bdevelopment.C76bb84e8, Build date: 2026-06-04 18:32:18 UTC
    Neighbor TTL: 121
    Local interface: ifp-0/1/47
    Local MAC address: 5c:07:58:79:0a:00
    Local port description: Physical interface #47 from node 0, chip 1
    Packets sent: 1
    Packets received: 12
    Neighbor status: Up
  Neighbor:
    Neighbor MAC address: 3c:fd:fe:d0:22:82
    Neighbor port ID: 3c:fd:fe:d0:22:82
    Neighbor TTL: 121
    Local interface: ifp-0/1/48
    Local MAC address: 5c:07:58:79:0a:00
    Local port description: Physical interface #48 from node 0, chip 1
    Packets sent: 1
    Packets received: 19
    Neighbor status: Up
  Neighbor:
    Neighbor MAC address: 3c:fd:fe:d0:22:80
    Neighbor port ID: 3c:fd:fe:d0:22:80
    Neighbor TTL: 121
    Local interface: ifp-0/1/49
    Local MAC address: 5c:07:58:79:0a:00
    Local port description: Physical interface #49 from node 0, chip 1
    Packets sent: 1
```

```
  Packets received: 19
  Neighbor status: Up
Neighbor: ufi19.q2a.u19.r6.nbg.rtbrick.net
  Neighbor MAC address: 5c:07:58:78:85:48
  Neighbor port ID: ifp-0/0/5
  Neighbor port description: Physical interface #5 from node 0, chip 0
  Neighbor system description: Rtbrick, Inc. s9510-28dc router, version 26.2.0-
g6daily.20260604183218+Bdevelopment.C76bb84e8, Build date: 2026-06-04 18:32:18 UTC
  Neighbor TTL: 121
  Local interface: ifp-0/1/5
  Local MAC address: 5c:07:58:79:0a:00
  Local port description: Physical interface #5 from node 0, chip 1
  Packets sent: 1
  Packets received: 1
  Neighbor status: Up
Neighbor: ufi19.q2a.u19.r6.nbg.rtbrick.net
  Neighbor MAC address: 5c:07:58:78:85:48
  Neighbor port ID: ifp-0/0/6
  Neighbor port description: Physical interface #6 from node 0, chip 0
  Neighbor system description: Rtbrick, Inc. s9510-28dc router, version 26.2.0-
g6daily.20260604183218+Bdevelopment.C76bb84e8, Build date: 2026-06-04 18:32:18 UTC
  Neighbor TTL: 121
  Local interface: ifp-0/1/6
  Local MAC address: 5c:07:58:79:0a:00
  Local port description: Physical interface #6 from node 0, chip 1
  Packets sent: 1
  Packets received: 1
  Neighbor status: Up
Neighbor: ufi20.q2a.u18.r6.nbg.rtbrick.net
  Neighbor MAC address: 5c:07:58:78:84:fc
  Neighbor port ID: ifp-0/0/7
  Neighbor port description: Physical interface #7 from node 0, chip 0
  Neighbor system description: Rtbrick, Inc. s9510-28dc router, version 26.2.0-
g6daily.20260604183218+Bdevelopment.C76bb84e8, Build date: 2026-06-04 18:32:18 UTC
  Neighbor TTL: 121
  Local interface: ifp-0/1/7
  Local MAC address: 5c:07:58:79:0a:00
  Local port description: Physical interface #7 from node 0, chip 1
  Packets sent: 1
  Packets received: 1
  Neighbor status: Up
Neighbor: ufi20.q2a.u18.r6.nbg.rtbrick.net
  Neighbor MAC address: 5c:07:58:78:84:fc
  Neighbor port ID: ifp-0/0/8
  Neighbor port description: Physical interface #8 from node 0, chip 0
  Neighbor system description: Rtbrick, Inc. s9510-28dc router, version 26.2.0-
g6daily.20260604183218+Bdevelopment.C76bb84e8, Build date: 2026-06-04 18:32:18 UTC
  Neighbor TTL: 121
  Local interface: ifp-0/1/8
  Local MAC address: 5c:07:58:79:0a:00
  Local port description: Physical interface #8 from node 0, chip 1
  Packets sent: 1
  Packets received: 1
  Neighbor status: Up
Neighbor: ufi18.q2c.u36.r6.nbg.rtbrick.net
  Neighbor MAC address: 5c:07:58:79:0f:00
  Neighbor port ID: ifp-0/1/96
  Neighbor port description: Physical interface #96 from node 0, chip 1
  Neighbor system description: Rtbrick, Inc. s9600-102xc router, version 26.2.0-
g6daily.20260604183222+Bdevelopment.C76bb84e8, Build date: 2026-06-04 18:32:22 UTC
  Neighbor TTL: 121
  Local interface: ifp-0/1/96
  Local MAC address: 5c:07:58:79:0a:00
  Local port description: Physical interface #96 from node 0, chip 1
  Packets sent: 1
  Packets received: 12
```

```
  Neighbor status: Up
Neighbor: ufi18.q2c.u36.r6.nbg.rtbrick.net
  Neighbor MAC address: 5c:07:58:79:0f:00
  Neighbor port ID: ifp-0/1/97
  Neighbor port description: Physical interface #97 from node 0, chip 1
  Neighbor system description: Rtbrick, Inc. s9600-102xc router, version 26.2.0-
g6daily.20260604183222+Bdevelopment.C76bb84e8, Build date: 2026-06-04 18:32:22 UTC
  Neighbor TTL: 121
  Local interface: ifp-0/1/97
  Local MAC address: 5c:07:58:79:0a:00
  Local port description: Physical interface #97 from node 0, chip 1
  Packets sent: 1
  Packets received: 12
  Neighbor status: Up
```

API To access the Operational State API that corresponds to this CLI, click include::common:partial$operational_api_links.adoc[tag=show_lldp_neighbor].

#### LLDP Summary

The command displays the LLDP global summary information.

**Syntax:**

**show lldp summary**

| Option | Description |
|---|---|
| - | Without any options, the command displays LLDP global summary details such as status, MAC address and description. |

Example: LLDP System Summary

```
supervisor@rtbrick: op> show lldp summary
Mode: global
  System hostname: ufi17.q2c.u34.r6.nbg.rtbrick.net
  System description: Rtbrick, Inc. s9600-102xc router, version 26.2.1-candidate.4, Build date: 2026-06-03 14:28:13 UTC
  Transmit interval: 30 sec
  Transmit holdtime: 120 sec
```

#### LLDP Error Statistics

The 'error statistics' option for `show lldp` command to display detailed transmit and receive error counters for LLDP operations.

**Syntax:**

**show lldp error statistics**

| Attribute | Description |
|---|---|
| `error statistics` | Displays detailed transmit and receive error counters for LLDP operations. |

The following example output displays a summary of LLDP transmit and receive error counters.

```
supervisor@rtbrick.net: op> show lldp error statistics
Sent Statistics:
  LLDP interface not present        : 0
  Ethernet header encode failed     : 0
  Packet Length exceeded            : 0

Received Statistics:
  Invalid destination mac           : 0
  Payload len error                 : 0
  LLDP chassis-id error             : 0
  Invalid chassis-id subtype        : 0
  LLDP port-id error                : 0
  Invalid port subtype              : 0
  LLDP ttl error                    : 0
  LLDP ttl len error                : 0
  Unsupported tlv                   : 0
  Unknown tlv                       : 0
  LLDP ttl len error                : 0
```

### LLDP Clear Commands

Clear commands allow to reset operational states.

#### LLDP Neighbor

This commands resets LLDP neighbor.

**Syntax:**

**clear lldp neighbor** <option>

| Option | Description |
|---|---|
| `all` | Clears all the LLDP neighbors. |
| `<ifp-0/0/1>` | Clears a specific neighbor on an interface. |

Example: The example below shows how to clear all the LLDP neighbors.

```
supervisor@rtbrick: op> clear lldp neighbor all
```

Example for the System Description TLV details. The `show lldp neighbor detail` command displays the neighbor's System Description TLV, providing device platform and other information learned via LLDP.

```
supervisor@rtbrick.net: op> show lldp neighbor detail
Neighbor: ufi23.q2c.u23.r7.nbg.rtbrick.net
  Neighbor MAC address: 5c:07:58:a2:cb:7d
  Neighbor port ID: ifp-0/1/63
  Neighbor port description: Physical interface #63 from node 0, chip 1
  Neighbor system description: Rtbrick, Inc. s9600-72xc router, version 26.2.1-alpha-
release.7, Build date: 2026-04-29 14:00:25
  Neighbor TTL: 121
  Local interface: ifp-0/1/62
  Local MAC address: 5c:07:58:a2:cb:7d
  Local port description: Physical interface #62 from node 0, chip 1
  Packets sent: 611
  Packets received: 610
  Neighbor status: Up
```

#### Clear LLDP

Clears the LLDP transmit and receive error statistics.

**Syntax:**

```
clear lldp error statistics
```

Example:

```
supervisor@rtbrick.net: op> clear lldp error statistics
Success: LLDP error statistics cleared
```

# 7. OAM

Ensure network reliability with integrated OAM tools that make monitoring simple, speed up fault detection, and streamline troubleshooting. RBFS OAM features deliver real-time visibility into network health, allowing operators to promptly verify connectivity, pinpoint failures, and maintain seamless service continuity across both Layer 2 and Layer 3.

Key Features:

- Connectivity Fault Management: Monitor and verify Ethernet service continuity.
- Bidirectional Forwarding Detection: Detect forwarding failures quickly for rapid convergence.
- Ping and Traceroute: Validate connectivity and diagnose network paths with built-in IP troubleshooting tools.

The following guides are available under the OAM section:

- Ethernet Connectivity Fault Management
- Bidirectional Forwarding Detection Overview
- Ping and Traceroute Commands

## 7.1. CFM

### 7.1.1. Ethernet Connectivity Fault Management

**Overview**

RBFS supports Connectivity Fault Management (CFM) that functions as an Ethernet layer Operations, Administration, and Maintenance (OAM) protocol. This feature enables connectivity monitoring, fault verification, and isolation, thereby ensuring network reliability.

CFM is used to detect, verify, and isolate connectivity issues within Ethernet networks. It operates over Ethernet, and utilizes Ethernet frames for communication.

RBFS OAM supports both point-to-point and point-to-multipoint CFM sessions.

**Nexthop Tracking**

RBFS supports nexthop tracking for Layer 2 issues using CFM (Connectivity Fault Management) sessions. It tracks the reachability of nexthop devices on a network path when CFM is enabled. The functionality enables detection of routing failures by continuously monitoring the availability and responsiveness of nexthop nodes.

CFM next-hop tracking is not supported for MPLS, VPN, IPv4 and IPv6 labeled unicast address families.

#### Supported Platforms

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

### 7.1.2. CFM Configuration

#### Configuring Connectivity Fault Management

To enable the connectivity fault management feature, you need to configure a CFM session.

##### Creating a CFM Session

OAM CFM supports both point-to-point and point-to-multipoint sessions.

**Syntax:**

```
set oam cfm session <session-name> <attribute> <value>
```

| Attribute | Description |
|---|---|
| `continuity-check-interval` | Specify continuity check interval time (in seconds). Supported intervals are: 1s, 10s and 100ms. Default value is 1 second. |
| `interface` | Logical interface name. |
| `level` | Specify the CFM level. Allowed range: 0 - 7. |
| `local-mep` | Specify the local end-point identifier. Range: 1-8191. |
| `maintenance-association` | Maintenance association name. |
| `maintenance-domain` | Maintenance domain name. |
| `remote-mep` | Remote end-point identifier. Range: 1-8191. |

API To access the RESTCONF API that corresponds to this CLI, click here.

Example: Point-to-point CFM session configuration

This following configuration defines two CFM sessions on two different interfaces. The sessions, named CFM_DEFAULT_LINK1 and CFM_DEFAULT_LINK2, use the same maintenance domains and different maintenance associations where the domain level is set to 2. The continuity-check interval is configured at 1 second. The continuity check messages (CCMs) are sent every second to verify the link connectivity.

The local MEP is specified as '1000' for CFM_DEFAULT_LINK1 and '1001' for CFM_DEFAULT_LINK2. The remote MEP values are set to '2000' for CFM_DEFAULT_LINK1 and '2001' for CFM_DEFAULT_LINK2 that indicates the remote endpoint for the link validation.

The local MEPs sends CCM messages every second to the remote MEPs. If no CCMs are received from the remote MEPs until CCM threshold (3.5 * CCM interval), it indicates a link fault and CFM session state is marked down.

```
set oam cfm session CFM_DEFAULT_LINK1
set oam cfm session CFM_DEFAULT_LINK1 level 2
set oam cfm session CFM_DEFAULT_LINK1 maintenance-domain MD_BNG
set oam cfm session CFM_DEFAULT_LINK1 maintenance-association MA_DEFAULT_100
set oam cfm session CFM_DEFAULT_LINK1 continuity-check-interval 1s
set oam cfm session CFM_DEFAULT_LINK1 local-mep 1000
set oam cfm session CFM_DEFAULT_LINK1 remote-mep 2000
set oam cfm session CFM_DEFAULT_LINK1 interface ifl-0/0/0/100
set oam cfm session CFM_DEFAULT_LINK2
set oam cfm session CFM_DEFAULT_LINK2 level 2
set oam cfm session CFM_DEFAULT_LINK2 maintenance-domain MD_BNG
set oam cfm session CFM_DEFAULT_LINK2 maintenance-association MA_DEFAULT_200
set oam cfm session CFM_DEFAULT_LINK2 continuity-check-interval 1s
set oam cfm session CFM_DEFAULT_LINK2 local-mep 1001
set oam cfm session CFM_DEFAULT_LINK2 remote-mep 2001
set oam cfm session CFM_DEFAULT_LINK2 interface ifl-0/0/1/200
```

```
supervisor@rtbrick.net: cfg> show config oam cfm session
{
  "rtbrick-config:session": [
    {
      "name": "CFM_DEFAULT_LINK1",
      "level": 2,
      "maintenance-domain": "MD_BNG",
      "maintenance-association": "MA_DEFAULT_100",
      "continuity-check-interval": "1s",
      "local-mep": 1000,
      "remote-mep": 2000,
      "interface": "ifl-0/0/0/100"
    },
    {
      "name": "CFM_DEFAULT_LINK2",
      "level": 2,
      "maintenance-domain": "MD_BNG",
      "maintenance-association": "MA_DEFAULT_200",
      "continuity-check-interval": "1s",
      "local-mep": 1001,
      "remote-mep": 2001,
      "interface": "ifl-0/0/1/200"
    }
  ]
}
```

To see related show commands and output, see Show OAM CFM.

Example 2: Point-to-Multipoint CFM Session configuration

The following is an example configuration that shows Point-to-Multipoint CFM sessions on a single

interface, that is ifl-0/0/5/0. With this configuration, the interface ifl-0/0/5/0 can run multiple CFM sessions. Each session is configured with a common local MEP (2000) and different remote MEPs (3000 and 4000). It allows the local endpoint to monitor connectivity with multiple remote endpoints on a single interface.

```
set oam cfm session CFM_R3
set oam cfm session CFM_R3 level 2
set oam cfm session CFM_R3 maintenance-domain BNG
set oam cfm session CFM_R3 maintenance-association MA_BNG
set oam cfm session CFM_R3 continuity-check-interval 1s
set oam cfm session CFM_R3 local-mep 2000
set oam cfm session CFM_R3 remote-mep 3000
set oam cfm session CFM_R3 interface ifl-0/0/5/0
set oam cfm session CFM_R4
set oam cfm session CFM_R4 level 2
set oam cfm session CFM_R4 maintenance-domain BNG
set oam cfm session CFM_R4 maintenance-association MA_BNG
set oam cfm session CFM_R4 continuity-check-interval 1s
set oam cfm session CFM_R4 local-mep 2000
set oam cfm session CFM_R4 remote-mep 4000
set oam cfm session CFM_R4 interface ifl-0/0/5/0
```

Example:

```
{
  "rtbrick-config:cfm": {
    "session": [
      {
        "name": "CFM_P2MP_R3",
        "level": 2,
        "maintenance-domain": "BNG",
        "maintenance-association": "MA_BNG",
        "continuity-check-interval": "1s",
        "local-mep": 2000,
        "remote-mep": 3000,
        "interface": "ifl-0/0/5/0"
      },
      {
        "name": "CFM_P2MP_R4",
        "level": 2,
        "maintenance-domain": "BNG",
        "maintenance-association": "MA_BNG",
        "continuity-check-interval": "1s",
        "local-mep": 2000,
        "remote-mep": 4000,
        "interface": "ifl-0/0/5/0"
      }
    ]
  }
```

```
}
```

To see related show commands and output, see Show OAM CFM.

#### Enabling Nexthop Tracking

You can enable Nexthop tracking using the following commands and options.

**Syntax:**

**set instance** <instance-name> **oam cfm track** <session-name> [nexthop4 | nexthop6] <ip-address>

| Attribute | Description |
|---|---|
| `<instance-name>` | Name of the instance. |
| `<session>` | Specify the session name. |
| `[nexthop4 \| nexthop6]` | Specify nexthop4 to trace ipv4 nexthop and specify nexthop6 to track ipv6 nexthop. |

API To access the RESTCONF API that corresponds to this CLI, click here.

Example:

The following configuration defines nexthop tracking using CFM sessions. The sessions, CFM_DEFAULT_LINK1 and CFM_DEFAULT_LINK2 are linked to a specific set of next-hop addresses, including both IPv4 and IPv6 addresses. If a CFM session goes down, all the associated next-hops will be considered as unreachable and the routes that uses these next-hops are made unavailable in the forwarding information base.

```
set instance default oam cfm track CFM_DEFAULT_LINK1
set instance default oam cfm track CFM_DEFAULT_LINK1 nexthop4 12.0.1.2
set instance default oam cfm track CFM_DEFAULT_LINK1 nexthop4 182.168.0.20
set instance default oam cfm track CFM_DEFAULT_LINK1 nexthop6 182:168::20
set instance default oam cfm track CFM_DEFAULT_LINK1 nexthop6 fe80::7810:2
set instance default oam cfm track CFM_DEFAULT_LINK2
set instance default oam cfm track CFM_DEFAULT_LINK2 nexthop4 12.1.1.2
set instance default oam cfm track CFM_DEFAULT_LINK2 nexthop4 182.168.0.20
set instance default oam cfm track CFM_DEFAULT_LINK2 nexthop6 182:168::20
set instance default oam cfm track CFM_DEFAULT_LINK2 nexthop6 fe80::7810:2
```

```
supervisor@rtbrick.net: cfg> show config instance default oam cfm
{
  "rtbrick-config:cfm": {
    "track": [
      {
        "session": "CFM_DEFAULT_LINK1",
```

```
        "nexthop4": [
          "12.0.1.2",
          "182.168.0.20"
          ],
        "nexthop6": [
          "182:168::20",
          "fe80::7810:2"
          ]
      },
      {
        "session": "CFM_DEFAULT_LINK2",
        "nexthop4": [
          "12.1.1.2",
          "182.168.0.20"
          ],
        "nexthop6": [
          "182:168::20",
          "fe80::7810:2"
          ]
      }
    ]
  }
}
```

#### CFM Ping Commands

RBFS supports CFM ping commands to test layer 2 connectivity to remote Maintenance End Points (MEPs) or remote MAC addresses. This command allows you to configure the maintenance domain, MA name, level, interface, packet count, and interval to troubleshoot and validate connectivity in CFM-enabled Ethernet networks.

**Syntax:**

**ping cfm remote-mac** <mac> **interface** <interface> <option>
**ping cfm remote-mep** [1 - 8191] **domain** <domain> **ma-name** <ma-name> <option>

- When using the `remote-mep` option, the `domain` and `ma-name` parameters are mandatory. All other parameters are optional.
- When using the `remote-mac` option, the `interface` parameter is mandatory. All other parameters are optional.

The following table provides command options and descriptions.

| Option | Description |
|---|---|
| `remote-mac <mac>` | Specifies the destination remote MAC address for the CFM ping operation. For `remote-mac`, the `interface` parameter is mandatory. All other parameters are optional. |

| Option | Description |
|---|---|
| `count [1 - 65535]` | Specifies the number of CFM ping packets to transmit. The default value is implementation-defined. Range 1 - 65535. |
| `interface <interface>` | Specifies the outgoing interface on which to send the CFM ping packets. |
| `interval [0.001 - 60.000]` | Specifies the interval, in seconds, between consecutive CFM ping packets. Range 0.001 - 60.000 seconds. |
| `level [0 - 7]` | Specifies the Maintenance Domain (MD) level to use for the CFM ping operation. Range 0 - 7. Default: 2. |
| `remote-mep` | Specifies the destination remote Maintenance End Point (MEP) ID for the CFM ping operation. For `remote-mep`, the `domain` and `ma-name` parameters are mandatory. All other parameters are optional. Range 1 - 8191. |
| `domain <domain>` | Specifies the Maintenance Domain (MD) associated with the remote MEP. |
| `ma-name <ma-name>` | Specifies the Maintenance Association (MA) name associated with the remote MEP. |

API To access the RESTCONF API that corresponds to this CLI, click here.

Example commands:

```
ping cfm remote-mac 7a:69:25:c0:00:02 interface ifl-0/0/0/0
ping cfm remote-mac 7a:69:25:c0:00:02 interface ifl-0/0/0/0 count 10 interval 0.1
ping cfm remote-mep 2000 domain MD_RBFS ma-name MA_OSPF count 20
ping cfm remote-mep 2000 domain MD_RBFS ma-name MA_OSPF interface ifl-0/0/0/0 count 10
interval 0.1
```

The following is an example for CFM ping using a remote MEP ID. It shows how to verify CFM connectivity to a remote Maintenance End Point by specifying the remote MEP ID, maintenance domain, and maintenance association.

```
supervisor@rtbrick.net: cfg> ping cfm remote-mep 2000 domain MD_RBFS ma-name MA_OSPF
cfm ping to 7a:69:25:c0:00:02
68 bytes from 7a:69:25:c0:00:02: seq=1 time=2.122 ms
68 bytes from 7a:69:25:c0:00:02: seq=2 time=1.083 ms
68 bytes from 7a:69:25:c0:00:02: seq=3 time=1.074 ms
68 bytes from 7a:69:25:c0:00:02: seq=4 time=1.097 ms
68 bytes from 7a:69:25:c0:00:02: seq=5 time=1.074 ms
--- cfm ping statistics ---
5 sent, 5 received, 0 errors, 0.00% loss, time 5000 ms
rtt min/avg/max/mdev = 1.074/1.289/2.122/0.432 ms
```

The following is an example for CFM ping using a remote MAC address. It shows how to verify CFM connectivity to a remote endpoint by specifying its MAC address and the outgoing interface.

```
supervisor@rtbrick.net: cfg> ping cfm remote-mac 7a:69:25:c0:00:02 interface ifl-
```

```
0/0/0/0
cfm ping to 7a:69:25:c0:00:02
68 bytes from 7a:69:25:c0:00:02: seq=1 time=2.135 ms
68 bytes from 7a:69:25:c0:00:02: seq=2 time=1.085 ms
68 bytes from 7a:69:25:c0:00:02: seq=3 time=2.151 ms
68 bytes from 7a:69:25:c0:00:02: seq=4 time=1.119 ms
68 bytes from 7a:69:25:c0:00:02: seq=5 time=1.075 ms
--- cfm ping statistics ---
5 sent, 5 received, 0 errors, 0.00% loss, time 5000 ms
rtt min/avg/max/mdev = 1.075/1.512/2.151/0.561 ms
```

### 7.1.3. Operational Commands

#### OAM CFM Show Commands

##### Show OAM CFM Session

The command displays the CFM session information.

**show oam cfm session** <option>

| Option | Description |
|---|---|
| `detail` | Shows detailed information about CFM sessions. |
| `interface` | Shows CFM session details for the specified logical interface. |

Example: CFM session

```
supervisor@rtbrick: op> show oam cfm session
Session                    Interface             Maintenance Domain    Maintenance
Association   Local MEP   Remote MEP  State
CFM_DEFAULT_LINK1          ifl-0/1/64/100        MD_BNG                MA_DEFAULT_100
1000        2000        Up
CFM_DEFAULT_LINK2          ifl-0/1/66/0          MD_BNG                MA_DEFAULT_200
1001        2001        Up
```

Example: CFM session details for a specific interface

```
supervisor@rtbrick: op> show oam cfm session interface ifl-0/1/64/100
Session name: CFM_DEFAULT_LINK1, Interface name: ifl-0/1/64/100, CFM state: Up
  Maintenance domain name: MD_BNG, Format: None
    Maintenance association name: MA_DEFAULT_100, Format: Character string
    Remote MAC address: e8:c5:7a:8f:55:b8
    Level: 2, Continuity check interval: 1s, Loss threshold: 3 Frames
    Local endpoint ID: 1000, Remote endpoint ID: 2000
    CCM packet received count: 9, Flap count: 0
    Last error: None
```

Example: Detailed view of CFM session

```
supervisor@rtbrick: op> show oam cfm session detail
Session name: CFM_DEFAULT_LINK1, Interface name: ifl-0/1/64/100, CFM state: Up
  Maintenance domain name: MD_BNG, Format: None
    Maintenance association name: MA_DEFAULT_100, Format: Character string
    Remote MAC address: e8:c5:7a:8f:55:b8
    Level: 2, Continuity check interval: 1s, Loss threshold: 3 Frames
    Local endpoint ID: 1000, Remote endpoint ID: 2000
    CCM packet received count: 10, Flap count: 0
    Last error: None
Session name: CFM_DEFAULT_LINK2, Interface name: ifl-0/1/66/0, CFM state: Up
  Maintenance domain name: MD_BNG, Format: None
    Maintenance association name: MA_DEFAULT_200, Format: Character string
    Remote MAC address: e8:c5:7a:8f:55:c0
    Level: 2, Continuity check interval: 1s, Loss threshold: 3 Frames
    Local endpoint ID: 1001, Remote endpoint ID: 2001
    CCM packet received count: 10, Flap count: 0
    Last error: None
```

Example for Point to Multipoint CFM sessions

```
supervisor@rtbrick: op> show oam cfm session
Session                   Interface             Maintenance Domain    Maintenance
Association   Local MEP   Remote MEP  State
CFM_R3                    ifl-0/1/27/0          BNG                   MA_BNG
3100        2100        Down
CFM_R4                    ifl-0/1/27/0          BNG                   MA_BNG
4100        2100        Down
```

Detailed view of Point to Multipoint CFM sessions.

```
supervisor@rtbrick: op> show oam cfm session detail
Session name: CFM_R3, Interface name: ifl-0/1/27/0, CFM state: Down
  Maintenance domain name: BNG, Format: None
    Maintenance association name: MA_BNG, Format: Character string
    Remote MAC address: 00:00:00:00:00:00
    Level: 2, Continuity check interval: 1s, Loss threshold: 3 Frames
    Local endpoint ID: 3100, Remote endpoint ID: 2100
    CCM packet received count: 0, Flap count: 0
    Last error: Interface down
Session name: CFM_R4, Interface name: ifl-0/1/27/0, CFM state: Down
  Maintenance domain name: BNG, Format: None
    Maintenance association name: MA_BNG, Format: Character string
    Remote MAC address: 00:00:00:00:00:00
    Level: 2, Continuity check interval: 1s, Loss threshold: 3 Frames
    Local endpoint ID: 4100, Remote endpoint ID: 2100
    CCM packet received count: 0, Flap count: 0
```

```
    Last error: Interface down
```

#### Show OAM CFM Statistics

This command displays OAM CFM statistics.

**show oam cfm statistics** <option>

| Option | Description |
|---|---|
| - | Without any option, the command shows CFM statistics for all interfaces. |
| interface | Shows CFM statistics for the specified logical interface. |

Example: CFM statistics for all interfaces

```
supervisor@rtbrick.net: cfg> show oam cfm statistics

Interface: ifl-0/0/0/1
  Received Statistics:
    Total packets                      : 1
    Invalid ethernet type              : 0
    Length error                       : 0
    CFM packets                        : 1
    Cross-connect CCM received         : 0
    Invalid CCM version                : 0
    RDI received                       : 0
    Continuity check interval mismatch : 0
    Invalid MD name format             : 0
    Invalid MA name format             : 0
  Sent Statistics:
    Total packets                      : 2
Interface: ifl-0/0/1/1
  Received Statistics:
    Total packets                      : 1
    Invalid ethernet type              : 0
    Length error                       : 0
    CFM packets                        : 1
    Cross-connect CCM received         : 0
    Invalid CCM version                : 0
    RDI received                       : 0
    Continuity check interval mismatch : 0
    Invalid MD name format             : 0
    Invalid MA name format             : 0
  Sent Statistics:
    Total packets                      : 2
```

Example: CFM statistics for a specific interface

```
supervisor@rtbrick.net: cfg> show oam cfm statistics interface ifl-0/0/0/1
Interface: ifl-0/0/0/1
  Received Statistics:
    Total packets                      : 205
    Invalid ethernet type              : 0
    Length error                       : 0
    CFM packets                        : 205
    Cross-connect CCM received         : 0
    Invalid CCM version                : 0
    RDI received                       : 0
    Continuity check interval mismatch : 0
    Invalid MD name format             : 0
    Invalid MA name format             : 0
  Sent Statistics:
    Total packets                      : 206
```

#### CFM Clear Commands

##### Clear OAM CFM Statistics

Clear commands allow to reset operational states.

**Syntax:**

**clear oam cfm statistics** <option>

| Option | Description |
|---|---|
| - | Clears all the CFM statistics. |
| <interface> | Clears statistics for the specified interface. |

Example commands:

```
clear oam cfm statistics
clear oam cfm statistics interface <interface>
```

# 7.2. BFD

## 7.2.1. Bidirectional Forwarding Detection Overview

BFD, or Bidirectional Forwarding Detection, is a fast failure detection protocol that enables RBFS devices to identify broken links quickly. This quick detection helps to ensure fast rerouting and minimizes downtime.

BFD swiftly identifies network failures between neighboring devices using a simple hello mechanism. BFD enabled RBFS devices establish a session and exchange BFD control packets at

regular intervals. If a device stops receiving BFD packets from its neighbor within the predefined detection time, it assumes that the path has failed. This failure information is then communicated to routing protocols such as OSPF, BGP, or IS-IS, allowing these protocols to reroute traffic immediately.

Routing protocols, such as OSPF and BGP, rely on hello and dead timers that can range from 1 to 10 seconds to detect failures. BFD ensures rapid convergence by detecting failures almost instantly and signaling routing protocols to switch paths quickly.

Imagine where Router A is connected to Router B. The routing protocol, such as OSPF, indicates that the neighbor is operational. However, due to a hardware fault, packets cannot actually be transmitted from Router A's interface to Router B. Without BFD, it could take several seconds for OSPF to recognize this problem. With BFD, the loss of BFD packets can signal a forwarding path failure swiftly.

This BFD implementation offers operational visibility and provides details such as session state (up/down/init), Timers (transmit interval, receive interval, detection time), Detection multiplier (how many missed packets before declaring down), packet statistics (sent/received counts) and last-down reason (for example, timeout, admin shutdown, remote fault).

Routing applications in RBFS can use BFD functionality through the OAMD infrastructure. OAM Daemon provides an interface that allows applications to create and delete BFD sessions.

#### Restrictions for Bidirectional Forwarding Detection

- Only 'Asynchronous' mode is supported. 'Demand' and 'Echo' modes are not supported.
- Currently, BFD sessions are supported on static routes, BGP, IS-IS, and OSPF protocols.
- Authentication is not supported.
- BFD sessions are limited only to single-hop.
- BFD currently supports only IPv4 and IPv6 address families, not MPLS.
- For a dual-stack IS-IS neighbor, it establishes two BFD sessions, one for IPv4 and another for IPv6 endpoints based the IS-IS configuration. If either BFD session goes down, the dual-stack IS-IS adjacency is immediately brought down.

#### Supported Platforms

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

### 7.2.2. BFD Configuration

#### Configuration Syntax and Commands

The following sections describe the Bidirectional Forwarding Detection configuration syntax and commands.

A static BFD session is configured by explicitly setting the local and peer IP addresses and interfaces along with the BFD parameters, such as 'transmit' and 'receive' intervals and 'detection multipliers'.

The session is established directly between the two endpoints without depending on any routing protocol.

In contrast, a dynamic BFD session is automatically created and managed by a routing protocol, such as BGP.

#### Static BFD Session Configuration

A BFD session can be established through static configuration.

**Syntax**

**set oam bfd session** <session name>

| Attribute | Description |
|---|---|
| `detect-multiplier` | The minimum number of consecutive BFD packets that must be missed before the BFD session is declared down. Range 2 - 255. Default 3. |
| `interface` | Logical interface name. |
| `local-v4-address` | Local IPv4 address. This is the IPv4 source address of BFD packets configured on the interface used in the session. |
| `local-v6-address` | Local IPv6 address. This is the IPv6 source address of BFD packets configured on the interface used in the session. |
| `peer-v4-address` | Peer IPv4 address. |
| `peer-v6-address` | Peer IPv6 address. |
| `receive-interval` | The receive interval is the amount of time (in milliseconds) that an RBFS device waits to receive a BFD control packet from its peer. Range: 100-10000. Default: 1000 milliseconds |
| `transmit-interval` | The transmit interval is the minimum amount of time (in milliseconds) that an RBFS device transmits BFD control packets to its peer. Range: 100 - 10000. Default: 1000 milliseconds. |

This configuration establishes two BFD sessions, named BFD_V4_IFP1_LINK1 and BFD_V6_IFP1_LINK1, for IPv4 and IPv6, respectively, on the same interface (ifl-0/0/0/0). Both sessions are set to a transmit/receive interval of 1000 milliseconds, meaning they exchange BFD packets once per second to quickly detect failures.

The peer IPv4 address, which is the neighbor's IP address for BFD packets, is specified as 192.0.2.2, while the local IPv4 address is designated as 192.0.2.1. For IPv6, the peer address is set as 2001:db8::2, and the local address is specified as 2001:db8::1.

If no packet is received within the hold interval (calculated as receive interval × detect multiplier), the BFD session is declared down.

```
set oam bfd session BFD_V4_IFP1_LINK1
set oam bfd session BFD_V4_IFP1_LINK1 peer-v4-address 192.0.2.2
set oam bfd session BFD_V4_IFP1_LINK1 local-v4-address 192.0.2.1
set oam bfd session BFD_V4_IFP1_LINK1 transmit-interval 1000
set oam bfd session BFD_V4_IFP1_LINK1 receive-interval 1000
set oam bfd session BFD_V4_IFP1_LINK1 interface ifl-0/0/0/0
set oam bfd session BFD_V6_IFP1_LINK1
set oam bfd session BFD_V6_IFP1_LINK1 peer-v6-address 2001:db8::2
set oam bfd session BFD_V6_IFP1_LINK1 local-v6-address 2001:db8::1
set oam bfd session BFD_V6_IFP1_LINK1 transmit-interval 1000
set oam bfd session BFD_V6_IFP1_LINK1 receive-interval 1000
set oam bfd session BFD_V6_IFP1_LINK1 interface ifl-0/0/0/0
```

```
{
  "rtbrick-config:session": [
    {
      "name": "BFD_V4_IFP1_LINK1",
      "peer-v4-address": "192.0.2.2",
      "local-v4-address": "192.0.2.1",
      "transmit-interval": 1000,
      "receive-interval": 1000,
      "interface": "ifl-0/0/0/0"
    },
    {
      "name": "BFD_V6_IFP1_LINK1",
      "peer-v6-address": "2001:db8::2",
      "local-v6-address": "2001:db8::1",
      "transmit-interval": 1000,
      "receive-interval": 1000,
      "interface": "ifl-0/0/0/0"
    }
  ]
}
```

**Enabling BFD for Static Routes**

RBFS supports tracking static routes through static BFD sessions. When a static BFD session moves to the down state, all the static routes associated with that session are automatically removed from the system. This ensures quick and reliable route failover.

To enable this feature, you must configure a static conditional profile with:

- `compare-type` : `bfd-session-up` (It means the condition is satisfied only when the BFD session is UP).
- `match-bfd-session` (the specific BFD session that you want to track).

After creating the conditional profile, attach it to the static routes that need monitoring. Once applied, these routes will remain active only while the corresponding static BFD session is

operational.

#### Static BFD Conditional Profile Configuration

**Syntax:**

Static Route Configuration with Conditional Profile:

> **set instance** <instance-name> **static conditional-profile** <conditional-profile-name> **compare-type bfd-session-up**
>
> **set instance** <instance-name> **static conditional-profile** <conditional-profile-name> **match-bfd-session** <bfd-session-name>

Linking Conditional Profile to Static Route:

> **set instance** <instance-name> **static route afi** <prefix> **safi** <safi> **nexthop-profile** <nexthop-profile-name> **conditional-profile** <conditional-profile-name>

| Attribute | Description |
|---|---|
| `<instance-name>` | Name of the instance. |
| `<nexthop-profile-name>` | Name of the next-hop profile. |
| `<conditional-profile-name>` | Name of the conditional profile. |
| `<bfd-session-name>` | Name of the BFD session. |

This configuration creates a conditional profile 'cp1' for static routes. The profile is used to track the status of a specific BFD session. The 'compare-type' is defined as `bfd-session-up`, it indicates the condition is satisfied only when the BFD session is UP. The `match-bfd-session` is specified as `BFD_V4_IFP1_LINK0`. Static routes that use this conditional profile will stay installed only when the BFD session 'BFD_V4_IFP1_LINK0' is UP. If the BFD session goes down, the routes are automatically withdrawn.

The route uses a next-hop profile np1 and is associated to the conditional profile 'cp1'. It means the route will be installed only if the condition in 'cp1' is met.

This configuration is used to ensure the static route is only installed when the monitored path is reachable. If the BFD session S1 goes down, the route is automatically withdrawn. It prevents traffic from being sent down an inactive path.

```
set instance default static route ipv4 192.0.2.26/32 unicast np1 conditional-profile
cp1
set instance default static nexthop-profile np1 nexthop 198.51.100.2
set instance default static conditional-profile cp1 compare-type bfd-session-up
set instance default static conditional-profile cp1 match-bfd-session
BFD_V4_IFP1_LINK0
```

```
{
  "rtbrick-config:static": {
    "route": {
      "ipv4": [
        {
          "prefix4": "192.0.2.26/32",
          "safi": "unicast",
          "nexthop-profile": "np1",
          "conditional-profile": [
            "cp1"
          ]
        }
      ]
    },
    "nexthop-profile": [
      {
        "name": "np1",
        "nexthop": "198.51.100.2"
      }
    ],
    "conditional-profile": [
      {
        "name": "cp1",
        "compare-type": "bfd-session-up",
        "match-bfd-session": "BFD_V4_IFP1_LINK0"
      }
    ]
  }
}
```

#### BFD Profile Configuration

A BFD profile contains the parameters that controls the behavior of BFD sessions. You can create BFD profiles that allow you to define common BFD parameters (such as transmit and receive intervals, detect multipliers) in the profile and associate them with the BGP, OSPF or IS-IS protocols.

### Syntax

**set oam bfd profile** <profile-name> <attribute> <value>

| Attribute | Description |
|---|---|
| `<profile name>` | Name of the BFD profile. |
| `detect-multiplier` | The minimum number of consecutive BFD packets that must be missed before the BFD session is declared down. Range 2 - 255. Default 3. |

| Attribute | Description |
|---|---|
| `receive-interval` | The receive interval is the minimum amount of time (in milliseconds) that an RBFS device waits to receive a BFD control packet from its peer. Range: 100 - 10000. Default: 1000 milliseconds |
| `transmit-interval` | The transmit interval is the minimum amount of time (in milliseconds) that an RBFS device transmits BFD control packets to its peer. Range: 100- 10000. Default: 1000 milliseconds. |

Example: BFD profile configuration

The following configuration creates a BFD profile named 'BGP_BFD_PROFILE1'. In this configuration, both the transmit interval and the receive interval are set to 1000 milliseconds, while the detect multiplier is defined as 3. It also defines the number of consecutive missed BFD packets required to declare the session as down.

```
set oam bfd profile BGP_BFD_PROFILE1
set oam bfd profile BGP_BFD_PROFILE1 transmit-interval 1000
set oam bfd profile BGP_BFD_PROFILE1 receive-interval 1000
set oam bfd profile BGP_BFD_PROFILE1 detect-multiplier 3
```

```
{
  "rtbrick-config:profile": [
    {
      "name": "BGP_BFD_PROFILE1",
      "transmit-interval": 1000,
      "receive-interval": 1000,
      "detect-multiplier": 3
    }
  ]
}
```

#### Attaching BFD Profile to BGP

After creating a BFD profile, you can attach the profile to BGP prot

### Syntax

**set instance** <instance-name> **protocol bgp peer interface** | **IPv4** | **IPv6** <interface name | local address & peer address> **bfd profile** <profile-name>

| Attribute | Description |
|---|---|
| `<profile name>` | Name of the BFD profile. |
| `interface` | Name of the interface. |

| Attribute | Description |
| --- | --- |
| `local address` | Local IPv4 and IPv6 address. |
| `peer address` | Peer IPv4 and IPv6 address. |

The following configuration attaches a BFD for different BGP peers in the default routing instance. The BFD profile named 'BGP_BFD_PROFILE1' is attached to the BGP peer interface 'ifl-0/0/0/6, IPv4 and IPv6 addresses (IPv4 local: 192.0.2.2 , peer: 192.0.2.1 and IPv6 local: 2001:db8::2, peer 2001:db8::1). This means that the BGP peer with these IP addresses will utilize the BFD parameters specified in BGP_BFD_PROFILE1. Any BGP peer that connects to this interface will automatically inherit the BFD settings. Once a BFD profile is attached to a peer group, all peers within that group will receive those BFD parameters.

```
set instance default protocol bgp peer interface ifl-0/0/0/6 bfd-profile
BGP_BFD_PROFILE1
set instance default protocol bgp peer ipv4 192.0.2.2 192.0.2.1 bfd-profile
BGP_BFD_PROFILE1
set instance default protocol bgp peer ipv6 2001:db8::2 2001:db8::1 bfd-profile
BGP_BFD_PROFILE1
```

```
{
  "rtbrick-config:peer": {
    "interface": [
      {
        "name": "ifl-0/0/0/6",
        "bfd-profile": "BGP_BFD_PROFILE1",
        "peer-group": "ipv6_bgp_ll"
      }
    ],
    "ipv4": [
      {
        "peer-address": "192.0.2.2",
        "update-source": "192.0.2.1",
        "bfd-profile": "BGP_BFD_PROFILE1",
        "peer-group": "ipv4_bgp"
      }
    ],
    "ipv6": [
      {
        "peer-address": "2001:db8::2",
        "update-source": "2001:db8::1",
        "bfd-profile": "BGP_BFD_PROFILE1",
        "peer-group": "ipv6_bgp"
      }
    ]
  }
}
```

#### Attaching BFD Profile on IS-IS Interface

Once the BFD profile is created, you can attach the BFD profile on IS-IS interface.

**Syntax**

**set instance** <instance-name> **protocol isis interface**<interface-name> **bfd-profile** <profile-name>

| Attribute | Description |
|---|---|
| `<instance-name>` | Name of the instance. |
| `interface-name` | Name of the interface. |
| `<profile-name>` | Name of the BFD profile. |

Example:

This configuration applies the BFD profile 'BFD_PROFILE_100MSx2' to the IS-IS interface 'ifl-0/0/0/0' so that IS-IS can detect adjacency link failures within 200 milliseconds.

```
set instance default protocol isis interface ifl-0/0/0/0
set instance default protocol isis interface ifl-0/0/0/0 bfd-profile
BFD_PROFILE_100MSx2
```

```
{
  "rtbrick-config:interface": [
    {
      "name": "ifl-0/0/0/0",
      "bfd-profile": "BFD_PROFILE_100MSx2"
    }
  ]
}
```

#### Enabling BFD for OSPFv2

Once the BFD profile is created, you can attach the BFD profile on the OSPFv2 interface.

**Syntax:**

**set instance** <instance-name> **protocol ospf address-family ipv4 area** <area-id> **interface** <interface-name> **bfd-profile** <profile-name>

| Attribute | Description |
|---|---|
| `<instance-name>` | Name of the instance. |

| Attribute | Description |
|---|---|
| `<area-id>` | OSPFv2 area identifier. |
| `<interface-name>` | Name of the interface. |
| `<profile-name>` | Name of the BFD profile. |

Example:

This configuration applies the BFD profile 'PROTOCOL_BFD_PROFILE5' to the OSPFv2 area 0.0.0.0 interface 'ifl-0/0/0/11', enabling OSPFv2 to detect neighbor link failures within 200 milliseconds.

```
set instance default protocol ospf address-family ipv4 area 192.0.2.0 interface ifl-
0/0/0/11
set instance default protocol ospf address-family ipv4 area 192.0.2.0 interface ifl-
0/0/0/11 bfd-profile PROTOCOL_BFD_PROFILE5
```

```
{
  "rtbrick-config:interface": [
    {
      "name": "ifl-0/0/0/11",
      "bfd-profile": "PROTOCOL_BFD_PROFILE5"
    }
  ]
}
```

#### Enabling BFD for OSPFv3

Once the BFD profile is created, you can attach the BFD profile on the OSPFv3 interface.

**Syntax:**

**set instance** <instance-name> **protocol ospf address-family ipv6 instance-id** <instance-id> **area** <area-id> **interface** <interface-name> **bfd-profile** <profile-name>

| Attribute | Description |
|---|---|
| `<instance-name>` | Name of the instance. |
| `<area-id>` | OSPFv2 area identifier. |
| `<instance-id>` | IPv6 instance identifier. |
| `<interface-name>` | Name of the interface. |
| `<profile-name>` | Name of the BFD profile. |

Example:

This configuration applies the BFD profile PROTOCOL_BFD_PROFILE5 to the OSPFv3 interface ifl-

0/0/0/11, enabling the interface to use the profile's defined BFD timers and detection parameters. It is configured under the IPv6 address family with instance-id 0 in area 0.0.0.0, ensuring OSPFv3 adjacencies on this interface use consistent BFD behavior.

```
set instance default protocol ospf address-family ipv6 instance-id 0 area 192.0.2.1
interface ifl-0/0/0/11
set instance default protocol ospf address-family ipv6 instance-id 0 area 192.0.2.1
interface ifl-0/0/0/11 bfd-profile PROTOCOL_BFD_PROFILE5
```

```
{
  "rtbrick-config:interface": [
    {
      "name": "ifl-0/0/0/11",
      "bfd-profile": "PROTOCOL_BFD_PROFILE5"
    }
  ]
}
```

#### Control Plane QoS Outbound Marking for BFD

You can configure Control Plane QoS outbound marking for BFD packets by specifying the DSCP value for BFD packets based on the routing protocol.

**Syntax:**

**set forwarding-options class-of-service control-plane-qos outbound-marking protocol** <protocol> <tos> ifp <ifp> codepoint <codepoint>

| Attribute | Description |
|---|---|
| <protocol> | Specifies the routing protocol for which to configure the DSCP marking for BFD packets. |
| <tos> | Specifies whether to use TOS for marking. |
| <codepoint> | Specifies the DSCP codepoint value to be used for marking BFD packets. |

Example:

The following command configures control-plane QoS so that all outgoing BFD packets sent by the device are remarked with a specific type of service (ToS) codepoint value 100.

```
set forwarding-options class-of-service control-plane-qos outbound-marking protocol
bfd tos codepoint 100
```

```
supervisor@rtbrick.net: cfg> show config forwarding-options class-of-service control-
plane-qos outbound-marking protocol bfd
{
  "rtbrick-config:protocol": [
    {
      "protocol": "bfd",
      "remark-type": "tos",
      "codepoint": 100
    }
  ]
}
```

### 7.2.3. BFD Operational Commands

#### BFD Show Commands

The command displays the BFD session information.

**show oam bfd session** <option>

| Option | Description |
|---|---|
| - | Without any option, it displays all the BFD session information. |
| history | BFD session history |
| peer | Peer address of BFD session |
| verbose | BFD session verbose. |
| session | Shows the session information. |
| detail | Shows detailed information about BFD sessions. |
| client | Shows information for specified client. |

Example: The output lists BGP neighbor relationships with each peer's address, remote AS number, state, uptime, and the number of prefixes exchanged.

```
supervisor@rtbrick: op> show bgp peer instance default
Instance: default
  Peer                                    Remote AS    State         Up/Down Time
PfxRcvd              PfxSent
  PE2                                     65542        Established   0d:00h:00m:45s
16                   24
  PE2                                     65542        Established   0d:00h:00m:46s
8                    24
  PE2                                     65542        Established   0d:00h:00m:40s
8                    24
```

Example: The output shows status of active BFD sessions with session state, and the transmit and receive intervals.

```
supervisor@rtbrick: op> show oam bfd session
Peer                          State      Transmit-interval(ms)  Receive-interval(ms)
Multiplier
192.0.2.2                     Up         1000                   1000
3
2001:db8:1:1::2               Up         1000                   1000
3
fe80::e8c5:7aff:fe8f:565f     Up         1000                   1000
3
```

Example: The output shows the IS-IS neighbor interface.

```
supervisor@rtbrick: op> show isis neighbor interface ifl-0/1/22/0
Instance: default
  Interface          System                   Level   State   Type       Up since
Expires
  ifl-0/1/22/0       1980.5110.0121.00        L1      Up      P2P        Fri Jun 05
12:56:44    in 29s 124973us
```

Example: This output shows the ISIS Level-1 adjacency on interface with the specified neighbor.

```
supervisor@rtbrick: op> show isis neighbor interface ifl-0/1/22/0 detail
Instance: default
  System: 1980.5110.0121.00, Interface: ifl-0/1/22/0
    State: Up, Level: L1, Adjacency type: P2P
    Circuit type: L1, Hold time: 30.0s, Expiry time: in 28s 13240us
    Local IPv4 address: 192.0.2.1, Remote IPv4 address: 192.0.2.2, Unnumbered: False
    Local IPv6 address: fe80::5c07:58ff:fe79:a17, Remote IPv6 address:
fe80::e8c5:7aff:fe8f:565f
    Up since: Fri Jun 05 12:56:44 GMT +0000 2026, Last down reason: NA
    Last transition: 2026-06-05T12:56:44.336339+0000, Number of transitions: 2
    Error counters:
      Level mismatch: 0, Area mismatch: 0, System ID: 0, Subnet mismatch: 0
      Hold timeout: 0, Neighbor down: 0, Interface down: 0, Admin reset: 0, BFD down:
0
      Interface configuration: 0, Area configuration: 0, Other: 0
```

Example: The output shows two BFD sessions (IPv4 and IPv6) status.

```
supervisor@rtbrick: op> show oam bfd session
Peer                          State      Transmit-interval(ms)  Receive-interval(ms)
Multiplier
192.0.2.2                     Up         1000                   1000
3
```

```
2001:db8:1:1::2                Up         1000                    1000
3
fe80::e8c5:7aff:fe8f:565f      Up         1000                    1000
3
```

Example: The output shows both IPv4 and IPv6 BFD sessions on the interface.

```
supervisor@rtbrick: op> show oam bfd session detail
Peer address: 192.0.2.2, Local address: 192.0.2.1
  State: Up, Interface: ifl-0/1/22/0, Instance : default
  Clients: isis
  Local discriminator: 1003, Remote discriminator: 1003
    Transmit interval: 1000ms, Receive interval: 1000ms, Detect time multiplier: 3
    Hold interval: 3000ms
    Diagnostic code: No Diagnostic
    Last transition time: 2026-06-05T12:57:28.589471+0000, Flap count: 1
Peer address: fe80::e8c5:7aff:fe8f:565f, Local address: fe80::5c07:58ff:fe79:a17
  State: Up, Interface: ifl-0/1/22/0, Instance : default
  Clients: isis
  Local discriminator: 1004, Remote discriminator: 1004
    Transmit interval: 1000ms, Receive interval: 1000ms, Detect time multiplier: 3
    Hold interval: 3000ms
    Diagnostic code: No Diagnostic
    Last transition time: 2026-06-05T12:57:28.589652+0000, Flap count: 1
```

Example output for OSPFv2 and OSPFv3 neighbor and the corresponding BFD session.

```
supervisor@rtbrick: op> show ospf neighbor interface ifl-0/1/22/100
Instance: default, Address family: ipv4
  Address                        Interface            Router          Area
State      Priority  DR              BDR             Uptime          Expires
  192.0.2.6                      ifl-0/1/22/100       198.51.100.21   0.0.0.0
Full       1         192.0.2.6       192.0.2.5       0d:00h:00m:27s  39s
Instance: default, Address family: ipv6, Instance ID: 0
  Address                        Interface            Router          Area
State      Priority  DR              BDR             Uptime          Expires
  fe80::e8c5:7a00:648f:565f      ifl-0/1/22/100       CBNG2           0.0.0.0
Full       1         198.51.100.21   198.51.100.11   0d:00h:00m:22s  39s
```

Example for BFD session associated with OSPFv2 neighbor.

```
supervisor@rtbrick: op> show oam bfd session peer 192.0.2.6
Peer                           State      Transmit-interval(ms)  Receive-interval(ms)
Multiplier
192.0.2.6                      Up         1000                    1000
3
```

Detailed view of BFD session associated with OSPFv2 neighbor.

```
supervisor@rtbrick: op> show oam bfd session peer 192.0.2.6 detail
Peer address: 192.0.2.6, Local address: 192.0.2.5
  State: Up, Interface: ifl-0/1/22/100, Instance : default
  Clients: ospf
  Local discriminator: 1005, Remote discriminator: 1005
    Transmit interval: 1000ms, Receive interval: 1000ms, Detect time multiplier: 3
    Hold interval: 3000ms
    Diagnostic code: No Diagnostic
    Last transition time: 2026-06-05T12:57:39.876497+0000, Flap count: 0
```

Example for BFD session associated with OSPFv3 neighbor.

```
supervisor@rtbrick: op> show oam bfd session peer fe80::e8c5:7a00:648f:565f
Peer                           State      Transmit-interval(ms)  Receive-interval(ms)
Multiplier
fe80::e8c5:7a00:648f:565f      Up         1000                   1000
3
```

Detailed view of BFD session associated with OSPFv3 neighbor.

```
supervisor@rtbrick: op> show oam bfd session peer fe80::e8c5:7a00:648f:565f detail
Peer address: fe80::e8c5:7a00:648f:565f, Local address: fe80::5c07:5800:6479:a17
  State: Up, Interface: ifl-0/1/22/100, Instance : default
  Clients: ospf
  Local discriminator: 1006, Remote discriminator: 1006
    Transmit interval: 1000ms, Receive interval: 1000ms, Detect time multiplier: 3
    Hold interval: 3000ms
    Diagnostic code: No Diagnostic
    Last transition time: 2026-06-05T12:57:39.876600+0000, Flap count: 0
```

The following example output shows that the BFD session with the peer.

```
supervisor@rtbrick: op> show oam bfd session peer 192.0.2.2 detail
Peer address: 192.0.2.2, Local address: 192.0.2.1
  State: Up, Interface: ifl-0/1/22/0, Instance : -
  Clients: static, Session name: BFD_V4_IFP1_LINK1
  Local discriminator: 1007, Remote discriminator: 1007
    Transmit interval: 1000ms, Receive interval: 1000ms, Detect time multiplier: 3
    Hold interval: 3000ms
    Diagnostic code: No Diagnostic
    Last transition time: 2026-06-05T12:58:01.784417+0000, Flap count: 0
```

The following example output shows that the static route that is installed.

```
supervisor@rtbrick: op> show route source static prefix 192.168.0.26/32
Flags: & - Imported
Instance: default, AFI: ipv4, SAFI: unicast
Flags  Prefix/Label                                Source               Pref  Next Hop
Outgoing Interface
       192.168.0.26/32                             static                  2
192.0.2.2                                  ifl-0/1/22/0
Instance: default, AFI: ipv4, SAFI: labeled-unicast
Flags  Prefix/Label                                Source               Pref  Next Hop
Label        Outgoing Interface
       192.168.0.26/32                             static                  2
192.0.2.2                      -           ifl-0/1/22/0
```

#### BFD Session Client

It allows filtering by client and provide summary, detailed, and history information.

**Syntax:**

**show oam bfd session client** <client>
**show oam bfd session client** <client> **detail**
**show oam bfd session client** <client> **verbose**
**show oam bfd session client** <client> **history**

The following table provides the command options and descriptions.

| Option | Description |
|---|---|
| session <client> | Displays BFD sessions associated with the specified client. |
| session <client> detail | Displays detailed information for BFD sessions associated with the specified client. |
| session <client> history | Displays historical state and event information for BFD sessions of the specified client. |
| session <client> verbose | Displays extended information for BFD sessions of the specified client. |

The following example displays detailed BFD session information for the specified client, including state, timers, diagnostics, and transition statistics.

```
supervisor@rtbrick.net: cfg> show oam bfd session client isis detail
Peer address: 12.0.7.2, Local address: 12.0.7.1
  State: Up, Interface: ifp-0/0/0/7, Instance : default
  Clients: isis
  Local discriminator: 1022, Remote discriminator: 1022
    Transmit interval: 100ms, Receive interval: 100ms, Detect time multiplier: 3
    Hold interval: 300ms
    Diagnostic code: No Diagnostic
```

```
    Last transition time: 2026-05-04T05:02:15.549548+0000, Flap count: 0
```

The following example displays extended BFD session information for the specified client, including timers, local and remote parameters, and so on.

```
supervisor@rtbrick.net: cfg> show oam bfd session client isis verbose
Peer address: 12.0.7.2, Local address: 12.0.7.1
  State: Up, Interface: ifp-0/0/0/7, Instance: default
  Clients: isis
  Local discriminator: 1022
    Configured transmit interval: 100ms, Configured receive interval: 100ms
    Desired minimum transmit interval: 100ms, Required minimum receive interval:
100ms, Detect time multiplier: 3
    Transmit interval: 91ms, Hold interval: 300ms
    Diagnostic code: No Diagnostic
  Remote discriminator: 1022, Remote state: Up
    Desired minimum transmit interval: 100ms, Required minimum receive interval:
100ms, Detect time multiplier: 3
    Diagnostic code: No Diagnostic
  Created time: 2026-05-04T05:02:15.212104+0000, Last transition time: 2026-05-
04T05:02:15.549548+0000
  Flap count: 0
```

The following example displays a summary list of BFD sessions for the specified client, including peer address, session state, and transmit/receive intervals.

```
supervisor@rtbrick.net: cfg> show oam bfd session client isis
Peer                          State      Transmit-interval(ms)  Receive-interval(ms)
12.0.7.2                      Up         100                    100
12.0.8.2                      Up         100                    100
12.1.7.2                      Up         100                    100
12.1.8.2                      Up         100                    100
fe80::7830:21                 Up         100                    100
fe80::7850:21                 Up         100                    100
fe80::8840:21                 Up         100                    100
fe80::8860:21                 Up         100                    100
```

#### BFD Clear Commands

This command resets the existing BFD sessions.

### Syntax

**clear oam bfd session** <option>

| Option | Description |
|---|---|
| `-` | Without any option, the commands clears all the BFD session details. |
| `session <session-name>` | Clears the specified BFD session. |
| `peer-address <peer-address> local-address <local-address>` | Clears BFD session for the specified peer. |

Example commands:

```
clear oam bfd session
clear oam bfd session peer-address <peer-address> local-address <local-address>
clear oam bfd session session-name <bfd-session-name>
```

# 7.3. Ping and Traceroute

## 7.3.1. Ping and Traceroute Commands

### Overview

RBFS provides the following ping and traceroute functionalities:

- IP Ping
- IP Ping over an MPLS Transport
- IP Traceroute
- IP Traceroute over an MPLS Transport
- MPLS LSP Ping
- MPLS LSP Traceroute

#### IP Ping

IP Ping is a tool that tests network reachability by sending ICMP echo requests. It provides information about the round-trip time of the packets and their status. PI Ping tests network reachability by sending ICMP echo requests. It provides information about the round-trip time (RTT) of the packets and their status. Ping is primarily used to verify general IP connectivity and diagnose basic network issues.

This utility performs a basic connectivity test using ICMP Echo Request and Reply messages. It checks the reachability of IP addresses between two endpoints, measures round-trip time, and assesses packet loss. The process works by sending ICMP Echo Requests from a source to a destination IP address. If the destination is reachable, it responds with Echo Reply messages.

#### IP Ping over an MPLS Transport

RBFS support ping utility that pings from one IP endpoint to another endpoint within an MPLS network. This ping checks the IP connectivity across an MPLS backbone and determines whether

MPLS forwarding allows the IP packet to reach its destination. The originating device send standard ICMP packets. These packets are label-switched within the MPLS core and are decapsulated at the egress edge router before reaching the destination. This method is used to validate Layer 3 VPN/IP reachability through an MPLS transport network.

#### IP Traceroute

IP Traceroute is a utility that is used to get information about the hop-by-hop path IP packets take between two devices. It identifies each router in the sequence and measures the latency for every hop. It can detect routing loops or points of packet loss. Traceroute operates by sending IP packets (using UDP, ICMP, or TCP) with progressively increasing Time-to-Live (TTL) values. Each router that decrements the TTL responds with a 'TTL exceeded' message, providing its address. Traceroute is useful for visualizing the exact route the traffic traverse across an IP network.

#### IP Traceroute over an MPLS Transport

IP Traceroute over an MPLS Transport allows you to trace an IP path between two endpoints when the underlying network uses MPLS for forwarding. The traceroute is initiated at the IP layer, the packets are transported through the MPLS core using label switching. It enables to understand how IP packets traverse an MPLS-based infrastructure and ensures that the IP reachability across the MPLS backbone. This utility allows to validate end-to-end connectivity across an MPLS core, especially in environments where IP traffic is tunneled or carried over label-switched paths.

It helps identify forwarding issues, pinpoint bottlenecks or failures within the MPLS transport, and verify correct label handling by transit routers. It provides visibility into the forwarding behavior of IP packets across the MPLS cloud, IP Traceroute over MPLS and enhances troubleshooting.

### MPLS LSP Ping

RBFS supports MPLS LSP (Label Switched Path) Ping that enables administrators to verify an LSP. LSP Ping works by sending packets through chosen LSP to a specific destination (remote endpoint) and checks for a response. It helps to identify if a Label Switched Path is disrupted.

A labeled packet (echo request) is sent towards a destination FEC to verify that this packet terminates at the correct egress LSR that services the FEC. If the egress node’s Echo Reply message is received, it confirms that the LSP is established successfully.

> MPLS LSP ping currently supports IS-IS for both IPv4 and IPv6 SID-Index, as well as LDP protocol for IPv4 and IPv6 FECs.

### MPLS LSP Traceroute

The MPLS LSP Traceroute utility is designed to verify the label switch path by sending MPLS Echo Request packets. It uses the Time-to-Live (TTL) mechanism to determine how many MPLS hops an MPLS packet goes through. This utility can map the route a packet takes to its destination and provides information about each hop, including the latency at each point.

As each Label Switching Router (LSR) along the path receives the packet, it decrements the MPLS TTL. When the TTL reaches 1, the router sends back an MPLS Echo Reply. The source router then sends additional MPLS Echo Request packets with TTL values incrementing from 2, 3, and so on.

Each time, a different LSR in the path responds as the TTL expires. This process continues until the egress LSR is reached.

> MPLS LSP traceroute supports IS-IS for both IPv4 and IPv6 SID-Index, as well as LDP for both IPv4 and IPv6 FECs.

#### Ping for CFM

RBFS supports CFM ping commands to test layer 2 connectivity to remote Maintenance End Points (MEPs) or remote MAC addresses.

For more information, see CFM Ping Commands.

### Supported Platforms

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

## 7.3.2. Ping and Traceroute Operational Commands

### IP Ping

The IP ping utility is used to test the reachability of a destination IP address and to measure the time taken for packets to travel between the source and the destination.

#### IP Ping in the Default Instance

The `ping` command allows you to ping to a destination to check if a device is reachable. When issued in the default instance, the command validates reachability using the default routing table.

### Syntax

```
ping <destination_ip> afi ipv4
ping <destination_ip> count [1 - 65535]
ping <destination_ip> do-not-fragment
ping <destination_ip> exit-interface <exit_interface>
ping <destination_ip> instance <instance>
ping <destination_ip> interval [0.001 - 60.000]
ping <destination_ip> safi [unicast | multicast | labeled-unicast]
ping <destination_ip> size [8 - 32768]
ping <destination_ip> source-interface <source>
ping <destination_ip> source-ip <source_ip>
ping <destination_ip> tos [0 - 255]
ping <destination_ip> ttl [1 - 255]
```

| | |
|---|---|
| `destination-ip` | **Specify the IPv4 destination IP address to ping.** |
| `interface` | Specify the source interface used to send the ping packets. |

| | |
|---|---|
| `destination-ip` | **Specify the IPv4 destination IP address to ping.** |
| `count` | Number of ping packets to send. (Default: 5) |
| `do-not-fragment` | Sets the `do-not-fragment` (DF) bit in the IP packet header. |
| `interval` | Time interval between packets in seconds. (Default: 1 second). |
| `safi` | Subsequent address family identifier. Supported options include 'unicast', 'multicast', and 'labeled-unicast'. Default is 'unicast'. |
| `size` | Size of the ping packet in bytes. (Default: 60 bytes). |
| `source-ip` | IPv4 address to be used as the source of the ping packets. |
| `ttl` | Time-to-live value applied to outgoing packets. Default value is 255. |
| `tos` | Type of Service field used for marking packets. Applicable to IP header only. Default value 0. |
| `exit-interface` | Enables to force ICMP echo requests through the specified exit interface if route is resolved via that interface. |

Example:

The following ping command sends ICMP echo requests (in the default instance) to a IPv4 destination address to verify reachability and measures round-trip latency. The output shows a response received from the destination that includes the `icmp_seq` (sequence number of the request), `ttl` (time-to-live value of the returned packet) and `time` (round-trip latency in milliseconds). The statistics summary displays the number of packets sent/received and the packet-loss percentage, indicating overall connectivity quality.

```
supervisor@rtbrick: op> ping 192.0.2.2
ping to 192.0.2.2 60 byte packets
68 bytes from 192.0.2.2: icmp_seq=1 ttl=64 time=29.73 ms
68 bytes from 192.0.2.2: icmp_seq=2 ttl=64 time=10.03 ms
68 bytes from 192.0.2.2: icmp_seq=3 ttl=64 time=10.00 ms
68 bytes from 192.0.2.2: icmp_seq=4 ttl=64 time=10.42 ms
68 bytes from 192.0.2.2: icmp_seq=5 ttl=64 time=9.97 ms
--- 192.0.2.2 ping statistics ---
5 sent, 5 received, 0 errors, 0.00% loss, time 5030.29 ms
rtt min/avg/max/mdev = 9.97/14.03/29.73/7.85 ms
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### IP Ping in a Specific Instance and AFI/SAFI

This command enables you to send ping requests to a destination within a selected routing instance or VRF, rather than the default global instance. You can also choose the required address family (IPv4 or IPv6). You can verify connectivity to check reachability issues in multi-VRF deployments.

### Syntax

**ping** <destination_ip> **afi ipv4**
**ping** <destination_ip> **count** <1 - 65535>
**ping** <destination_ip> **do-not-fragment**
**ping** <destination_ip> **instance** <instance>
**ping** <destination_ip> **interval** <0.001 - 60.000>
**ping** <destination_ip> **safi** [**unicast** | **multicast** | **labeled-unicast**]
**ping** <destination_ip> **size** <8 - 32768>
**ping** <destination_ip> **source-interface** <source>
**ping** <destination_ip> **source-ip** <source_ip>
**ping** <destination_ip> **tos** <0 - 255>
**ping** <destination_ip> **ttl** <1 - 255>

| `destination-ip` | **IPv4 destination address to ping.** |
|---|---|
| `instance-name` | Name of the instance in which the ping must be executed. |
| `afi` | Address Family Identifier. |
| `safi` | Subsequent address family identifier. Supported options include 'unicast', 'multicast', and 'labeled-unicast'. Default is 'unicast' |
| `interface` | Source interface used to send the ping packets. |
| `count` | Number of ping packets to transmit. (Default: 5). |
| `do-not-fragment` | Set the `do-not-fragment` (DF) bit in IP packet header. |
| `interval` | Time interval between packets in seconds. (Default: 1 second). |
| `size` | Size of the ping packet in bytes. (Default: 60 bytes). |
| `source-ip` | IPv4 address to be used as the source of the ping packets. |
| `ttl` | Time-to-live value applied to outgoing packets. Default value is 255. |
| `tos` | Type of Service field used for marking packets. Applicable to IP header only. Default value is 0. |

AFI and SAFI are optional attributes. If they are not explicitly specified, the system defaults to AFI 'IPv6' and SAFI 'unicast'.

Example:

The following example shows how to ping the destination address within the routing instance vrf1.

```
supervisor@rtbrick: op> ping 198.51.100.21 instance default afi ipv4 safi labeled-
unicast
ping to 198.51.100.21 60 byte packets
68 bytes from 198.51.100.21: icmp_seq=1 ttl=64 time=9.94 ms
68 bytes from 198.51.100.21: icmp_seq=2 ttl=64 time=9.97 ms
68 bytes from 198.51.100.21: icmp_seq=3 ttl=64 time=9.98 ms
68 bytes from 198.51.100.21: icmp_seq=4 ttl=64 time=10.19 ms
```

```
68 bytes from 198.51.100.21: icmp_seq=5 ttl=64 time=9.97 ms
--- 198.51.100.21 ping statistics ---
5 sent, 5 received, 0 errors, 0.00% loss, time 5027.39 ms
rtt min/avg/max/mdev = 9.94/10.01/10.19/0.09 ms
```

API To access the RESTCONF API that corresponds to this CLI, click here.

### IPv6 Ping

The IPv6 ping utility is used to verify the reachability of an IPv6 destination by sending ICMPv6 echo requests. It helps to check IPv6 connectivity and measure basic round-trip delay. It is useful for troubleshooting IPv6 networks.

#### IPv6 Ping in the Default Instance

The ping command allows you to ping to an IPv6 destination to check an IPv6 destination in the default routing instance. It helps to verify whether a device is reachable over IPv6 and provides basic round-trip time and connectivity information.

**Syntax**:

**ping** <destination_ipv6> **afi ipv6**
**ping** <destination_ipv6> **count** <1-65535>
**ping** <destination_ipv6> **instance** <instance>
**ping** <destination_ipv6> **interval** <0.001-60.000>
**ping** <destination_ipv6> **safi** <unicast | multicast | labeled-unicast>
**ping** <destination_ipv6> **size** <8 - 32768>
**ping** <destination_ipv6> **source-interface** <source>
**ping** <destination_ipv6> **source-ip** <source_ip>
**ping** <destination_ipv6> **tos** <0-255>
**ping** <destination_ipv6> **ttl** <1-255>

| | |
|---|---|
| `destination-ipv6` | **IPv6 destination address.** |
| `interface` | Source interface to use for sending packets. |
| `count` | Number of ping packets to send (Default: 5). |
| `do-not-fragment` | Sets the 'do-not-fragment' (DF) bit in IP packet header. |
| `interval` | Interval time between the successive packets (Default: 1 second). |
| `size` | Size of the ping packet in bytes (Default: 60 bytes). |
| `source-ipv6` | IPv6 source address to use for the ping. |
| `ttl` | Time-to-Live value for the packet. Default value is 255. |
| `tos` | Type of Service (TOS) value to use in the packet. Applicable to IP header only. Default value 0 |

Example:

The following ping command sends ICMP echo requests (in the default instance) to a IPv6 destination address to verify reachability and measures round-trip latency.

```
supervisor@rtbrick: op> ping 2001:db8:1:1::2
ping to 2001:db8:1:1::2 60 byte packets
68 bytes from 2001:db8:1:1::2: icmp_seq=1 ttl=64 time=10.15 ms
68 bytes from 2001:db8:1:1::2: icmp_seq=2 ttl=64 time=9.94 ms
68 bytes from 2001:db8:1:1::2: icmp_seq=3 ttl=64 time=9.98 ms
68 bytes from 2001:db8:1:1::2: icmp_seq=4 ttl=64 time=10.09 ms
68 bytes from 2001:db8:1:1::2: icmp_seq=5 ttl=64 time=10.00 ms
--- 2001:db8:1:1::2 ping statistics ---
5 sent, 5 received, 0 errors, 0.00% loss, time 5029.07 ms
rtt min/avg/max/mdev = 9.94/10.03/10.15/0.07 ms
```

#### IPv6 Ping in a Specific Instance and AFI/SAFI

This functionality allows you to ping an IPv6 destination within a selected routing instance (VRF). You can specify the IPv6 address family to ensure the ping is executed using the correct protocol. It helps to validate IPv6 connectivity for services operating outside the default instance and to troubleshoot reachability in multi-VRF environments.

### Syntax

**ping** <destination_ipv6> **afi ipv6**
**ping** <destination_ipv6> **count** <1-65535>
**ping** <destination_ipv6> **instance** <instance>
**ping** <destination_ipv6> **interval** <0.001-60.000>
**ping** <destination_ipv6> **safi** <unicast | multicast | labeled-unicast>
**ping** <destination_ipv6> **size** <8 - 32768>
**ping** <destination_ipv6> **source-interface** <source>
**ping** <destination_ipv6> **source-ip** <source_ip>
**ping** <destination_ipv6> **tos** <0-255>
**ping** <destination_ipv6> **ttl** <1-255>

| `destination-ipv6` | **IPv6 destination address.** |
|---|---|
| `instance-name` | Name of the instance on which ping must be executed. |
| `afi` | IPv4 Address Family Identifier. |
| `safi` | Subsequent address family identifier. Supported options include unicast, multicast, and labeled-unicast. Default is unicast. |
| `interface` | Source interface to use for sending packets. |
| `count` | Number of ping packets to send (Default: 5). |
| `do-not-fragment` | Sets the `do-not-fragment` (DF) bit in IP packet header. |
| `interval` | Interval time between successive packets (Default: 1 second). |

| | |
|---|---|
| destination-ipv6 | **IPv6 destination address.** |
| size | Size of the ping packet (Default: 60). |
| source-ipv6 | IPv6 source address. |
| ttl | Time-to-live value to be used in the packet. Default value is 255. |
| tos | Type of Service (ToS) value to use in packet. Applicable to IP header only. Default value 0. |

AFI and SAFI are optional attributes. If they are not explicitly specified, the system defaults to AFI 'IPv6' and SAFI 'unicast'.

Example:

In the following example, the command sends ICMP echo requests to the IPv6 destination address. It verifies IPv6 connectivity over an MPLS labeled-unicast path within VRF and the output confirms successful or failure scenario.

```
supervisor@rtbrick: op> ping 2001:db8:1:1::2
ping to 2001:db8:1:1::2 60 byte packets
68 bytes from 2001:db8:1:1::2: icmp_seq=1 ttl=64 time=10.15 ms
68 bytes from 2001:db8:1:1::2: icmp_seq=2 ttl=64 time=9.94 ms
68 bytes from 2001:db8:1:1::2: icmp_seq=3 ttl=64 time=9.98 ms
68 bytes from 2001:db8:1:1::2: icmp_seq=4 ttl=64 time=10.09 ms
68 bytes from 2001:db8:1:1::2: icmp_seq=5 ttl=64 time=10.00 ms
--- 2001:db8:1:1::2 ping statistics ---
5 sent, 5 received, 0 errors, 0.00% loss, time 5029.07 ms
rtt min/avg/max/mdev = 9.94/10.03/10.15/0.07 ms
```

### IP Traceroute

#### IP Traceroute in the Default Instance

This command traces the path taken by packets to a specified IP destination using the default routing instance.

**Syntax:**

**traceroute** <destination_ip> **afi** <ipv4 | ipv6>
**traceroute** <destination_ip> **instance** <instance>
**traceroute** <destination_ip> **interval** <0.001 - 60.000>
**traceroute** <destination_ip> **maxhop** <1 - 255>
**traceroute** <destination_ip> **repeat** <1 - 65535>
**traceroute** <destination_ip> **safi** <unicast | multicast | labeled-unicast>
**traceroute** <destination_ip> **size** <1 - 65535>
**traceroute** <destination_ip> **source-interface** <source>
**traceroute** <destination_ip> **source-ip** <source_ip>

| | |
|---|---|
| destination-ip | **IPv4 destination address.** |
| interface | Name of the source interface. |
| repeat | Number of probe packets sent per hop (Default: 3). |
| safi | Subsequent address family identifier. Supported options include unicast, multicast, and labeled-unicast. Default is unicast. |
| interval | Time interval between probe packets (Default: 1 second). |
| size | Size of each traceroute packet in bytes (Default: 60). |
| source-ip | Source IP address . |
| maxhop | Maximum number of hops before the TTL expires (Default: 30). |

Example:

This following traceroute command output shows a traceroute to the destination using the default settings. The trace completes in two hops and displays the intermediate router at 198.51.100.90 followed by the destination, along with the round-trip time for three probe packets at each hop.

```
supervisor@rtbrick: op> traceroute 192.0.2.2
traceroute to 192.0.2.2 30 hops max, 60 byte packets
1    192.0.2.2    9.91ms    10.00ms    9.93ms
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### IP Traceroute in Specific Instance and AFI/SAFI

This command performs a traceroute to a specified IP destination using the routing table of a specific VRF.

**Syntax:**

**traceroute** <destination_ip> **afi** <ipv4 | ipv6>
**traceroute** <destination_ip> **instance** <instance>
**traceroute** <destination_ip> **interval** <0.001 - 60.000>
**traceroute** <destination_ip> **maxhop** <1 - 255>
**traceroute** <destination_ip> **repeat** <1 - 65535>
**traceroute** <destination_ip> **safi** <unicast | multicast | labeled-unicast>
**traceroute** <destination_ip> **size** <1 - 65535>
**traceroute** <destination_ip> **source-interface** <source>
**traceroute** <destination_ip> **source-ip** <source_ip>

| **destination-ip** | **IPv4 destination address.** |
|---|---|
| instance-name | Name of the routing instance in which the traceroute is executed. |
| afi | Specifies the IP address family to use. |

| destination-ip | IPv4 destination address. |
|---|---|
| safi | Subsequent address family identifier. Supported options include unicast, multicast, and labeled-unicast. Default is unicast. |
| interface | Name of the source interface used to send traceroute probe packets. |
| repeat | Number of probe packets sent to each hop. (Default: 3). |
| interval | Time interval between packets in seconds. (Default: 1 second). |
| size | Size of the probe packet in bytes. (Default: 60 bytes). |
| source-ip | Source IP address for the traceroute probes. |
| maxhop | Maximum number of hops before the TTL expires. (Default: 30). |

AFI and SAFI are optional attributes. If they are not explicitly specified, the system defaults to AFI 'IPv6' and SAFI 'unicast'.

In the following example, the traceroute command verifies IPv4 MPLS LSP connectivity (AFI IPv4, SAFI labeled-unicast) from the specified source interface/IP. The output shows label-switched forwarding and successful reachability of the destination 198.51.100.80 in the next hop. It indicates correct MPLS label distribution and forwarding along the path.

```
supervisor@rtbrick: op> traceroute 198.51.100.21 instance default afi ipv4 safi
labeled-unicast source-interface ifl-0/1/22/100 source-ip 192.0.2.5
traceroute to 198.51.100.21 30 hops max, 60 byte packets
1    198.51.100.21    9.95ms    10.01ms    9.95ms
```

For the MPLS transport, it displays the ingress labels in each hop.

### IPv6 Traceroute

#### IPv6 Traceroute in the Default Instance

This command allows you to traceroute the hop-by-hop path to a given IP destination.

### Syntax

**traceroute** <destination_ipv6> **afi** [**ipv4** | **ipv6**]
**traceroute** <destination_ipv6> **instance** <instance>
**traceroute** <destination_ipv6> **interval** [0.001 - 60.000]
**traceroute** <destination_ipv6> **maxhop** [1 - 255]
**traceroute** <destination_ipv6> **repeat** [1 - 65535]
**traceroute** <destination_ipv6> **safi** [**unicast** | **multicast** | **labeled-unicast**]
**traceroute** <destination_ipv6> **size** [1 - 65535]
**traceroute** <destination_ipv6> **source-interface** <source>
**traceroute** <destination_ipv6> **source-ip** <source_ip>

| destination-ipv6 | IPv6 destination address. |
|---|---|
| interface | Source interface name. |
| repeat | Number of packets sent per hop (Default: 3) |
| interval | Interval time between packets, in seconds (Default: 1) |
| size | Size of the traceroute packet in bytes (Default: 60) |
| source-ip | IPv6 source address. |
| maxhop | Maximum number of hops before the TTL expires (Default: 30). |

The following example shows the IPv6 traceroute traces the hop-by-hop path to IPv6 destination. It shows the successful reachability within two hops and reporting round-trip latency for each hop using default traceroute parameters.

```
supervisor@rtbrick: op> traceroute 2001:db8:1:1::2
traceroute to 2001:db8:1:1::2 30 hops max, 60 byte packets
1    2001:db8:1:1::2    9.94ms    10.52ms    9.96ms
```

#### IPv6 Traceroute in a Specific Instance and AFI/SAFI

This command allows you to traceroute to a particular IPv6 destination in a specific VRF.

### Syntax

**traceroute** <destination_ipv6> **afi** [ipv4 | ipv6]
**traceroute** <destination_ipv6> **instance** <instance>
**traceroute** <destination_ipv6> **interval** [0.001 - 60.000]
**traceroute** <destination_ipv6> **maxhop** [1 - 255]
**traceroute** <destination_ipv6> **repeat** [1 - 65535]
**traceroute** <destination_ipv6> **safi** [unicast | multicast | labeled-unicast]
**traceroute** <destination_ipv6> **size** [1 - 65535]
**traceroute** <destination_ipv6> **source-interface** <source>
**traceroute** <destination_ipv6> **source-ip** <source_ip>

| destination-ipv6 | IPv6 destination address. |
|---|---|
| instance-name | Instance on which the traceroute command must be executed. |
| afi | Address Family Identifier (AFI). |
| safi | Subsequent address family identifier. Supported options include unicast, multicast, and labeled-unicast. Default is unicast. |
| interface | Name of the source interface. |
| repeat | Number of packets for each hop (Default, 3). |
| interval | The interval period between the packets (Default, 1 second). |
| size | Size of the traceroute packet (Default, 60). |

| destination-ipv6 | IPv6 destination address. |
|---|---|
| `source-ip` | Source IPv6 address. |
| `maxhop` | Maximum number of hops before the TTL expires (Default, 30) |

The AFI and SAFI attributes are optional. If not specified, the default AFI is 'IPv6' and the default SAFI 'unicast'.

The following example traceroute verifies IPv6 reachability using MPLS labeled-unicast in the default routing instance. The probe is sourced from a specific interface. The first hop is an MPLS hop (with the label).

```
supervisor@rtbrick: op> traceroute 2001:db8:0:1::21 instance default afi ipv6 safi
labeled-unicast source-interface ifl-0/1/22/100 source-ip 2001:db8:2:1::1
traceroute to 2001:db8:0:1::21 30 hops max, 60 byte packets
1    2001:db8:0:1::21    9.97ms    10.05ms    9.95ms
```

For MPLS transport paths, the output displays the ingress MPLS label at each hop.

**LSP Ping**

The `ping mpls` command allows you to ping to an MPLS labelled switch path destination to see if a networked device is reachable.

**Syntax**:

**ping mpls** <destination_ip_fec> **fec-type** [ **ldp** | **isis** ]
**ping mpls** <destination_ip_fec> **instance** <instance>
**ping mpls** <destination_ip_fec> **maxhop** [1 - 64]
**ping mpls** <destination_ip_fec> **repeat** [1 - 10]
**ping mpls** <destination_ip_fec> **size** [1 - 8600]
**ping mpls** <destination_ip_fec> **source-ip** <source_ip>
**ping mpls** <destination_ip_fec> **wait-time** [1 - 60]

| Attribute | Description |
|---|---|
| `fec-type` | Specifies the type of FEC to help the device identify the exact type. Default 'fec-type' is LDP. |
| `instance` | Specify the instance name. |
| `wait-time` | Specify the interval time (in seconds) between successive ping packets to send. |
| `size` | The size of the PAD TLV to be included in the MPLS Echo Request packet. (Default: 60 bytes). |

| Attribute | Description |
|---|---|
| `repeat` | Total number ping attempts the device sends to the destination. Default 3. |
| `maxhop` | Maximum number of hops the packet traverse (Default: 30). |
| `source-ip` | Specify the source IP address for the MPLS Echo Request. |

Example: LSP ping for a prefix

In the following example, the MPLS ping verifies LDP-based MPLS connectivity to the FEC 62.154.10.70/32 by sending MPLS echo requests along the label-switched path. All probes are successfully replied. The output shows 0 percent packet loss and stable round-trip times. The RTTs range from 7.61 ms to 47.30 ms confirms that the MPLS LSP to the destination is reachable and operational.

```
supervisor@rtbrick: op> ping mpls 198.51.100.22/32
Sending 5 MPLS echoes to ldp FEC 198.51.100.22/32
Success: Response from 192.0.2.2: Egress for the FEC: seq = 1 rtt = 39.30 ms
Success: Response from 192.0.2.2: Egress for the FEC: seq = 2 rtt = 2.26 ms
Success: Response from 192.0.2.2: Egress for the FEC: seq = 3 rtt = 2.22 ms
Success: Response from 192.0.2.2: Egress for the FEC: seq = 4 rtt = 2.22 ms
Success: Response from 192.0.2.2: Egress for the FEC: seq = 5 rtt = 2.33 ms
--- 198.51.100.22/32 mpls ping statistics ---
5 sent, 5 received, 0 errors, 0.00% loss, time 5103.94 ms
rtt min/avg/max = 2.22/9.67/39.30 ms
```

Example: LSP ping for a FEC type

In the following example, the MPLS ping tests MPLS reachability to the FEC 62.154.20.70/32 using ISIS FEC. MPLS echo requests are sent along the IS-IS–signaled label-switched path. Four replies are received from 217.239.37.58. It is identified as the egress LSR for this FEC. While one probe does not receive a response, resulting in 20% packet loss. The measured RTTs range from 11.23 ms to 34.01 ms. It indicates that the MPLS LSP is reachable, though with an intermittent loss observed.

```
supervisor@rtbrick: op> ping mpls 198.51.100.22/32 fec-type ldp
Sending 5 MPLS echoes to ldp FEC 198.51.100.22/32
Success: Response from 192.0.2.2: Egress for the FEC: seq = 1 rtt = 2.27 ms
Success: Response from 192.0.2.2: Egress for the FEC: seq = 2 rtt = 2.36 ms
Success: Response from 192.0.2.2: Egress for the FEC: seq = 3 rtt = 2.36 ms
Success: Response from 192.0.2.2: Egress for the FEC: seq = 4 rtt = 2.21 ms
Success: Response from 192.0.2.2: Egress for the FEC: seq = 5 rtt = 1.14 ms
--- 198.51.100.22/32 mpls ping statistics ---
5 sent, 5 received, 0 errors, 0.00% loss, time 5104.11 ms
rtt min/avg/max = 1.14/2.07/2.36 ms
```

API To access the RESTCONF API that corresponds to this CLI, click here.

### MPLS LSP Traceroute

The `traceroute mpls` command is used to trace the path of an MPLS Label Switched Path (LSP) hop by hop.

**Syntax**:

**traceroute mpls** <destination_ipv6_fec> **fec-type** [**ldp** | **isis**]
**traceroute mpls** <destination_ipv6_fec> **instance** <instance>
**traceroute mpls** <destination_ipv6_fec> **maxhop** [1 - 64]
**traceroute mpls** <destination_ipv6_fec> **repeat** [1 - 10]
**traceroute mpls** <destination_ipv6_fec> **size** [1 - 8600]
**traceroute mpls** <destination_ipv6_fec> **source-ip** <source_ip>
**traceroute mpls** <destination_ipv6_fec> **wait-time** [1 - 60]

| Attribute | Description |
|---|---|
| `-` | Without any option, the command traceroute for the specified prefix. |
| `fec-type [ldp \| isis]` | Specifies the type of FEC to help the device identify the exact type. Default 'fec-type' is LDP. |
| `instance` | Specify the instance name. Instance in which the prefix needs to be looked up (Default: 'default') |
| `maxhop` | Maximum number of hops (Default: 30). |
| `size` | The size of the PAD TLV to be included in the MPLS Echo Request packet. (Default: 60 bytes). |
| `wait-time` | Specify the wait time (in seconds) to receive the response. (1-60). |
| `repeat` | Number of packets to be repeated with same TTL. Default 3. |
| `source-ip` | Source IP address. |

Example: LSP traceroute for a FEC type

The following example shows the MPLS traceroute traces the LDP-signaled label-switched path to the FEC 192.168.0.40/32 by sending MPLS echo requests with increasing TTL values. The output shows three MPLS hops in the LSP. At each transit hop, labels are swapped. It confirms correct LDP label forwarding. The final hop (34.0.0.4) is identified as the egress LSR for the FEC. All probes are successfully received (0% loss), with measured RTTs at each hop.

```
Sending MPLS echoes to ldp FEC 192.168.0.40/32 for traceroute
TTL  Address       Label    Protocol  Down Addr                Down Label  Status
RTT
  1  12.0.0.2      519942   ldp       23.0.0.3                 519938      Label
switched at stack-depth 1    45.15ms, 3.20ms, 3.21ms
  2  23.0.0.3      519938   ldp       34.0.0.4                 3           Label
switched at stack-depth 1    3.18ms, 4.25ms, 2.17ms
```

```
  3  34.0.0.4      3        ldp       -                        -           Egress for
the FEC                 37.18ms, 4.23ms, 3.22ms
--- 192.168.0.40/32 mpls traceroute statistics ---
9 sent, 9 received, 0 errors, 0.00% loss, time 17617.15 ms
```

API To access the RESTCONF API that corresponds to this CLI, click here.

### Show Commands

#### Show OAM MPLS

The command displays LSP ping command details.

**Syntax:**

**show oam mpls** <option>

| Option | Description |
|---|---|
| `statistics` | Displays MPLS LSP ping statistics. |

Example: OAM MPLS LSP Ping statistics

The `show oam mpls statistics` output indicates that no MPLS OAM activity has occurred on the device. No MPLS echo requests or replies have been sent or received. All transmit and receive error counters remain at zero. It indicates that MPLS OAM is currently idle, with no tests.

```
supervisor@rtbrick: op> show oam mpls statistics
Sent Statistics:
MPLS echo request sent          : 13
MPLS echo reply sent            : 0
Sent error                      : 0
Sent error - No table           : 0
Sent error - No FEC             : 0
Sent error - No Adjacency       : 0
Sent error - No MPLS Adjacency  : 0
Sent error - No interface       : 0
Sent error - Encode failure     : 0

Received Statistics:
MPLS echo received              : 0
MPLS echo request received      : 0
MPLS echo reply received        : 13
Receive error - L3 decode       : 0
Receive error - header decode   : 0
Receive error - TLV decode      : 0
Receive error - FEV TLV decode  : 0
Receive error - unsupported FEV : 0
```

```
Receive error - no session      : 0
```

#### Show OAM IP Statistics

The 'show oam ip statistics' command displays information about sent and received OAM IP packet counters.

**Syntax:**

```
show oam ip statistics
```

The following example show command output displays the sent and received statistics for an echo request message.

```
supervisor@rtbrick: cfg> show oam ip statistics
Sent Statistics:
Echo request sent                          : 0
Sent error                                 : 0
Sent error - no table                      : 0
Sent error - no route                      : 2
Sent error - no adjacency                  : 0
Sent error - no interface                  : 0
Sent error - encode failure                : 0

Received Statistics:
Echo reply received                        : 0
Echo reply - TTL expired                   : 0
Echo reply - destination unreachable       : 0
Receive error - L3 decode                  : 0
Receive error - no session                 : 0
```

### Clear Commands

#### clear oam ip statistics

The 'clear oam ip statistics' command is used to reset the OAM IP statistics counters.

**Syntax**:

```
clear oam ip statistics
```

Example for 'clear oam ip statistics' command:

```
supervisor@rtbrick: cfg> clear oam ip statistics
Success: IP oam statistics cleared
```

# 8. Multicast

The Multicast section of the RtBrick FullStack (RBFS) User Guides provides configuration and operational details for essential multicast protocols and features.

Key Features:

- Details how to manage local group memberships using the Internet Group Management Protocol (IGMP).
- Explains how to build and maintain traffic distribution trees across the network via Protocol Independent Multicast (PIM).
- Focuses on configuring Multicast VPN services to securely transport multicast traffic over a shared IP or MPLS core infrastructure.

The following guides available under the Multicast section:

- IGMP
- PIM
- Multicast VPN

## 8.1. IGMP

### 8.1.1. IGMP Overview

Internet Group Management (IGMP) protocol allows a host to advertise its multicast group membership to neighboring switches and routers. IGMP is a standard protocol used by the TCP/IP protocol suite to achieve dynamic multicasting.

There are two components in the IGMP solution:

- IGMPv2/v3 Client: It sends Join or Leave messages to a multicast group. Typical example of a client is a SET-TOP box. The IGMP client can respond to any IGMP general queries or group-specific queries that are received.
- Multicast Router: The recipient of IGMP Join/Leave message. After receiving the message, it determines whether the corresponding message needs to be processed or not. After processing the IGMP messages, it sends this information to its multicast upstream router. Along with this, it can program certain entries in its routers which results in forwarding specific multicast packets on that interface.

#### IGMPv3 Lite

IGMP version 3 adds support for "source filtering", that is, the ability for a system to report interest in receiving packets **only** from specific source addresses, or from **all but** specific source addresses, sent to a particular multicast address. That information may be used by multicast routing protocols to avoid delivering multicast packets from specific sources to networks where there are no interested receivers.

The RtBrick IGMP v3lite solution adds support for source filtering. Source filtering enables a multicast receiver host to signal from which groups it wants to receive multicast traffic, and from which sources this traffic is expected. That information may be used by multicast routing protocols to avoid delivering multicast packets from specific sources to networks where there are no interested receivers.

IGMP Version 3 will help conserve bandwidth by allowing a host to select the specific sources from which it wants to receive traffic. Also, multicast routing protocols will be able to make use of this information to conserve bandwidth when constructing the branches of their multicast delivery trees.

#### Static Joins

After an interface on a multicast device is configured to statically join an IGMP group, the multicast device considers that the interface has static multicast group members and sends multicast packets to this interface, regardless of whether hosts connected to this interface request the multicast packets.

#### SSM Mapping

SSM mapping takes IGMPv2 reports and converts them to IGMPv3. In case of legacy devices, there could be a possibility that BNG might receive IGMPv2 membership reports. If BNG receives an IGMPv2 membership for a specific group G1, BNG uses the SSM mapping configuration to determine one or more Source (S) addresses for a given group. This SSM mappings are translated to the IGMPv3 joins like IGMPV3 JOIN INCLUDE (G, [S1, G1], [S2, G1] and so on) and BNG continues to process as if it has received from the subscriber.

### Supported Platforms

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

## 8.1.2. IGMP Configuration

### Configuration Hierarchy

The diagram illustrates the IGMP configuration hierarchy.

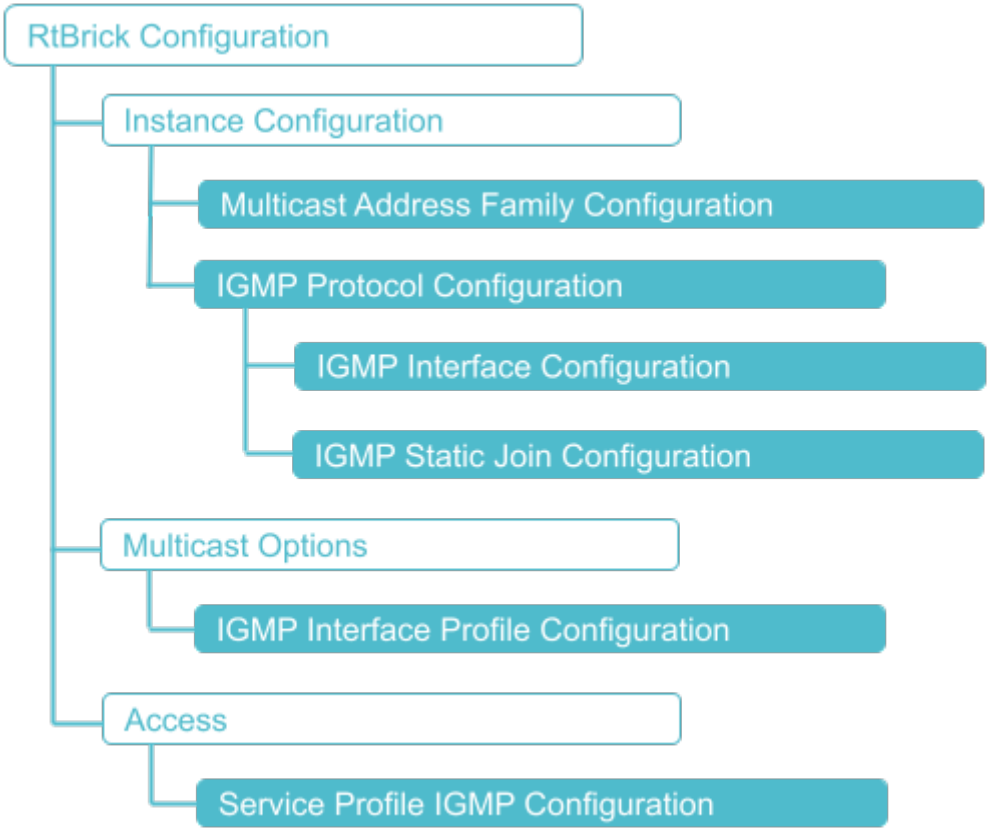



*Figure 147. IGMP Configuration Hierarchy*

### Configuration Syntax and Commands

The following sections describe the interface configuration syntax and commands.

#### Multicast Address Family Configuration

You can enable the multicast IPv4 address family under the IGMP instance using the following command:

**Syntax:**

**set instance** <instance> **address-family** <attribute> <value>

| Attribute | Description |
|---|---|
| `<instance>` | Specifies the name of the network instance |
| `<afi>` | Address family identifier (AFI). Supported value: ipv4 |
| `<safi>` | Subsequent address family identifier (SAFI), that is, multicast. |

Example: Multicast Address Family Configuration

```
{
    "rtbrick-config:address-family": [
      {
```

```
        "afi": "ipv4",
        "safi": "multicast"
      }
    ]
  }
```

#### IGMP Protocol Configuration

To configure an IGMP on an instance, the same instance should be enabled globally with AFI IPv4 and SAFI as both unicast and multicast.

**Syntax:**

**set instance** <instance> **protocol igmp** <attribute> <values>

If no instance is specified, IGMP will be enabled on the default instance.

| Options | Description |
|---|---|
| `<instance>` | Name of the IGMP instance. |
| `interfaces <…>` | IGMP interface configuration. Refer to the IGMP Interface Configuration section for the IGMP interface configuration. |
| `robustness-variable <variable-value>` | The robustness value is used by IGMP to determine the number of times to send messages. Default value: 3. Range: 0-255. |
| `source-address <source-address>` | Source address of the IGMP query at the instance-level.<br>NOTE: IF subscriber IFL is configured with the source address, then takes priority; otherwise, the the instance-level source address will be used. If source address is not configured, 0.0.0.0 will be the default address. |
| `static-group <…>` | Static multicast route configuration. Refer to the IGMP Static Join Configuration section for details on the IGMP static join configuration. |

Example: IGMP Configuration

```
{
    "rtbrick-config:igmp": {
      "robustness-variable": 5,
      "source-address": "198.51.100.177"
    }
  }
```

To access the RESTCONF API that corresponds to this CLI, click here.

#### IGMP Interface Configuration

When you start IGMP on an interface, it operates with the default settings.

**Syntax:**

**set instance** <instance> **protocol igmp interfaces interface** <interface-name> <attribute> <value>

| Attribute | Description |
|---|---|
| `<instance>` | Name of the instance |
| `<interface-name>` | Name of the IP multicast interface |
| `max-groups <count>` | Specifies the maximum count of multicast group memberships |
| `version <version>` | Specifies the IGMP version, that is, IGMPv2 or IGMPv3 |
| `interface-profile <profile>` | Name of the interface configuration profile |

Example: IGMP Interface Configuration

```
{
    "rtbrick-config:interface": [
      {
        "interface-name": "ifl-0/0/0/1",
        "version": "IGMPv3",
        "max-groups": 30,
        "interface-profile": "profile1"
      }
    ]
  }
```

To access the RESTCONF API that corresponds to this CLI, click here.

#### IGMP Static Join Configuration

**Syntax:**

**set instance** <instance> **protocol igmp static-group** <attribute> <value>

**Command Parameters**

| | |
|---|---|
| `<instance>` | Specifies the instance name |
| `<group-address>` | Specifies the multicast address |

| | |
|---|---|
| `<outgoing-interface>` | Name of the outbound interface. The `null0` is a discard or sink interface for IGMP static join configuration. If IGMP Static Join is configured with the Null0 outgoing interface, then it is mandatory to configure the null0 IGMP interface. |

Example: IGMP Static Join Configuration

```
{
    "rtbrick-config:static-group": [
      {
        "group-address": "198.51.100.200",
        "source-address": "198.51.100.1",
        "outgoing-interface": "null0"
      }
    ]
  }
```

To access the RESTCONF API that corresponds to this CLI, click here.

#### IGMP Interface Profile Configuration

**Syntax:**

**set multicast-options igmp interface-profile** <attribute> <value>

| Attribute | Description |
|---|---|
| `filter-policy <filter-policy>` | Specifies the filter policy. The policy should be defined under policy statement. |
| `immediate-leave <enable \| disable>` | Enable or disable the immediate leave option. The `immediate-leave` attribute removes group membership immediately upon receiving a group leave membership report. If enabled, IGMP perform an immediate leave upon receiving an IGMP group leave message. If the router is IGMP-enabled, it sends an IGMP last member query with a last member query response time. However, the router does not wait for the response time before it prunes off the group querier-timeout-interval IGMP other querier timeout. Default: 425s |
| `query-interval <query-interval>` | IGMP query interval in seconds. The query interval ranges from 1 to 1024 seconds. The default value is 125 seconds. |
| `query-max-response-time <query-max-response-time>` | Maximum query response interval in seconds. The maximum query response interval ranges from 1 to 1024 seconds. The default value is 100 seconds. |
| `ssm-map-policy <ssm-map-policy>` | IGMP SSM policy name. The policy for (,G) **mapping to (S,G)** |

| Attribute | Description |
|---|---|
| `start-query-count <start-query-count>` | Specifies the number of queries sent out on startup, separated by the Start Query Interval. The start query count ranges from 1 to 1024. The default value is 3. |
| `start-query-interval <start-query-interval>` | Specifies the start query interval. The start-query-interval ranges from 1 to 1024 seconds. The default value is 31 seconds (query-interval/4). |

Example: IGMP Interface Profile Configuration

```
{
    "rtbrick-config:interface-profile": [
      {
        "profile-name": "profile1",
        "immediate-leave": "enable",
        "query-interval": 30,
        "query-max-response-time": 10,
        "start-query-count": 10,
        "start-query-interval": 10,
        "filter-policy": "filter_policy",
        "ssm-map-policy": "ssm_policy"
      }
    ]
  }
```

#### Service Profile IGMP Configuration

**Syntax:**

**set access service-profile** <profile-name> **igmp** <attribute> <value>

| Attribute | Description |
|---|---|
| `<profile-name>` | Name of the service profile |
| `enable <true\|false>` | Enable IGMP service |
| `max-members <max-members>` | Maximum IGMP membership per subscriber |
| `profile <profile>` | IGMP profile |
| `version [IGMPv1/IGMPv2/IGMPv3]` | IGMP version. The default IGMP version is IGMPv3. |

Example: Service Profile IGMP Configuration

```
{
    "rtbrick-config:service-profile": [
      {
        "profile-name": "service-profile1",
```

```
        "igmp": {
          "enable": "true",
          "profile": "INTERFACE_PROFILE_1",
          "version": "IGMPv3",
          "max-members": 10
        }
      }
    ]
  }
```

**IGMP Configuration Example**

```
{
  "ietf-restconf:data": {
    "rtbrick-config:instance": [
      {
        "name": "default",
        "protocol": {
          "igmp": {
            "robustness-variable": 5,
            "source-address": "198.51.100.91"
          }
        }
      }
    ],
    "rtbrick-config:multicast-options": {
      "igmp": {
        "interface-profile": [
          {
            "profile-name": "INTERFACE_PROFILE_1",
            "query-interval": 10,
            "filter-policy": "FILTER_POLICY_1"
          },
          {
            "profile-name": "INTERFACE_PROFILE_2",
            "query-interval": 20,
            "ssm-map-policy": "SSM_POLICY_1"
          }
        ]
      }
    },
    "rtbrick-config:policy": {
      "statement": [
        {
          "name": "FILTER_POLICY_1",
          "ordinal": [
            {
              "ordinal": 1,
              "match": {
                "rule": [
```

```
                {
                  "rule": 1,
                  "type": "ipv4-mcast-group",
                  "value-type": "discrete",
                  "match-type": "or-longer",
                  "value": "198.51.100.20/24"
                }
              ]
            },
            "action": {
              "rule": [
                {
                  "rule": 1,
                  "operation": "return-deny"
                }
              ]
            }
          },
          {
            "ordinal": 2,
            "action": {
              "rule": [
                {
                  "rule": 1,
                  "operation": "return-permit"
                }
              ]
            }
          }
        ]
      },
      {
        "name": "SSM_POLICY_1",
        "ordinal": [
          {
            "ordinal": 1,
            "match": {
              "rule": [
                {
                  "rule": 1,
                  "type": "ipv4-mcast-group",
                  "value-type": "discrete",
                  "match-type": "or-longer",
                  "value": "198.51.100.10/24"
                }
              ]
            },
            "action": {
              "rule": [
                {
                  "rule": 1,
```

```
                    "type": "ipv4-mcast-source",
                    "operation": "overwrite",
                    "value": "198.51.100.11/24"
                  }
                ]
              }
            }
          ]
        }
      ]
    },
    "rtbrick-config:access": {
      "interface": {
        "double-tagged": [
          {
            "interface-name": "ifp-0/0/1",
            "outer-vlan-min": 1,
            "outer-vlan-max": 4049,
            "inner-vlan-min": 1,
            "inner-vlan-max": 4049,
            "access-type": "PPPoE",
            "access-profile-name": "access-profile1",
            "service-profile-name": "service-profile1",
            "aaa-profile-name": "aaa-profile1"
          }
        ]
      },
      "service-profile": [
        {
          "profile-name": "service-profile1",
          "igmp": {
            "enable": "true",
            "profile": "INTERFACE_PROFILE_1",
            "version": "IGMPv3",
            "max-members": 10
          }
        }
      ]
    }
  }
}
```


### 8.1.3. IGMP Operational Commands

**IGMP Show Commands**

**Syntax:**

**show igmp** <option>

| Option | Description |
|---|---|
| `group` | IGMP group summary information |
| `group <group>` | IGMP group detailed information |
| `group instance <name>` | IGMP group summary information in a specific instance |
| `group outgoing-interface <interface_name>` | IGMP group detailed information over a specific interface |
| `interface` | IGMP logical-interface summary information |
| `interface instance <name>` | IGMP interface summary information on specific instance |
| `interface <interface_name>` | IGMP interface detailed information |

Example: Display IGMP interface details for all instances

```
supervisor@rtbrick: op> show igmp interface
Interface                       Primary Address   State          Querier Address
Instance       Uptime
ppp-0/1/23/72339069014638629    0.0.0.0           Querier        198.51.100.11
default        00h:00m:03s
ppp-0/1/23/72339069014638630    0.0.0.0           Querier        198.51.100.11
default        00h:00m:03s
ppp-0/1/23/72339069014638631    0.0.0.0           Querier        198.51.100.11
default        00h:00m:03s
ppp-0/1/23/72339069014638632    0.0.0.0           Querier        198.51.100.11
default        00h:00m:03s
ppp-0/1/23/72339069014638633    0.0.0.0           Querier        198.51.100.11
default        00h:00m:03s
```

Example: Display the interface summary for a specific instance

```
supervisor@rtbrick: op> show igmp interface instance default
Interface                       Primary Address   State          Querier Address
Instance       Uptime
ppp-0/1/23/72339069014638629    0.0.0.0           Querier        198.51.100.11
default        00h:00m:04s
ppp-0/1/23/72339069014638630    0.0.0.0           Querier        198.51.100.11
default        00h:00m:04s
ppp-0/1/23/72339069014638631    0.0.0.0           Querier        198.51.100.11
default        00h:00m:04s
ppp-0/1/23/72339069014638632    0.0.0.0           Querier        198.51.100.11
default        00h:00m:04s
ppp-0/1/23/72339069014638633    0.0.0.0           Querier        198.51.100.11
default        00h:00m:04s
```

Example: Display IGMP group summary on all instances

```
supervisor@rtbrick: op> show igmp group
```

```
Source Address      Group Address       Interface                       Instance
Uptime         Expires      Version
192.0.2.54          232.252.0.1         null0                           default
00h:01m:17s    n/a          IGMP
192.0.2.54          232.252.0.2         null0                           default
00h:01m:17s    n/a          IGMP
192.0.2.54          232.252.0.1         ppp-0/1/23/72339069014638629    default
00h:00m:02s    3m 44s       IGMPv3
192.0.2.54          232.252.0.1         ppp-0/1/23/72339069014638630    default
00h:00m:02s    3m 44s       IGMPv3
192.0.2.54          232.252.0.1         ppp-0/1/23/72339069014638631    default
00h:00m:02s    3m 44s       IGMPv3
192.0.2.54          232.252.0.1         ppp-0/1/23/72339069014638632    default
00h:00m:02s    3m 44s       IGMPv3
192.0.2.54          232.252.0.1         ppp-0/1/23/72339069014638633    default
00h:00m:02s    3m 44s       IGMPv3
```

Example: Display the group summary on specific instance

```
supervisor@rtbrick: op> show igmp group instance default
Source Address      Group Address       Interface                       Instance
Uptime         Expires      Version
192.0.2.54          232.252.0.1         null0                           default
00h:01m:19s    n/a          IGMP
192.0.2.54          232.252.0.2         null0                           default
00h:01m:19s    n/a          IGMP
192.0.2.54          232.252.0.1         ppp-0/1/23/72339069014638629    default
00h:00m:04s    3m 42s       IGMPv3
192.0.2.54          232.252.0.1         ppp-0/1/23/72339069014638630    default
00h:00m:04s    3m 42s       IGMPv3
192.0.2.54          232.252.0.1         ppp-0/1/23/72339069014638631    default
00h:00m:04s    3m 42s       IGMPv3
192.0.2.54          232.252.0.1         ppp-0/1/23/72339069014638632    default
00h:00m:04s    3m 42s       IGMPv3
192.0.2.54          232.252.0.1         ppp-0/1/23/72339069014638633    default
00h:00m:04s    3m 42s       IGMPv3
```

Example: Display detailed group information for specific group and source on all instances

```
supervisor@rtbrick: op> show igmp group 232.252.0.1 192.0.2.54
(192.0.2.54, 232.252.0.1)
  Outgoing interface      : null0
  Instance                : default
  Source                  : Static
  State                   : No Members Present
  Version                 : IGMP
  Uptime                  : 00h:01m:20s
  Expires                 : n/a
  Membership interval     : n/a
```

```
  Last reporter           : n/a
  Last member query count : n/a
  Last member interval    : n/a
  Retransmit time         : n/a
  Max response time       : n/a
(192.0.2.54, 232.252.0.1)
  Outgoing interface      : ppp-0/1/23/72339069014638629
  Instance                : default
  Source                  : Dynamic
  State                   : Members Present
  Version                 : IGMPv3
  Uptime                  : 00h:00m:05s
  Expires                 : 3m 41s
  Membership interval     : 225s
  Last reporter           : 198.51.100.101
  Last member query count : 3
  Last member interval    : 1s
  Retransmit time         : 1s
  Max response time       : 0s
(192.0.2.54, 232.252.0.1)
  Outgoing interface      : ppp-0/1/23/72339069014638630
  Instance                : default
  Source                  : Dynamic
  State                   : Members Present
  Version                 : IGMPv3
  Uptime                  : 00h:00m:05s
  Expires                 : 3m 41s
  Membership interval     : 225s
  Last reporter           : 198.51.100.102
  Last member query count : 3
  Last member interval    : 1s
  Retransmit time         : 1s
  Max response time       : 0s
(192.0.2.54, 232.252.0.1)
  Outgoing interface      : ppp-0/1/23/72339069014638631
  Instance                : default
  Source                  : Dynamic
  State                   : Members Present
  Version                 : IGMPv3
  Uptime                  : 00h:00m:05s
  Expires                 : 3m 41s
  Membership interval     : 225s
  Last reporter           : 198.51.100.103
  Last member query count : 3
  Last member interval    : 1s
  Retransmit time         : 1s
  Max response time       : 0s
(192.0.2.54, 232.252.0.1)
  Outgoing interface      : ppp-0/1/23/72339069014638632
  Instance                : default
  Source                  : Dynamic
```

```
  State                   : Members Present
  Version                 : IGMPv3
  Uptime                  : 00h:00m:05s
  Expires                 : 3m 41s
  Membership interval     : 225s
  Last reporter           : 198.51.100.104
  Last member query count : 3
  Last member interval    : 1s
  Retransmit time         : 1s
  Max response time       : 0s
(192.0.2.54, 232.252.0.1)
  Outgoing interface      : ppp-0/1/23/72339069014638633
  Instance                : default
  Source                  : Dynamic
  State                   : Members Present
  Version                 : IGMPv3
  Uptime                  : 00h:00m:05s
  Expires                 : 3m 41s
  Membership interval     : 225s
  Last reporter           : 198.51.100.105
  Last member query count : 3
  Last member interval    : 1s
  Retransmit time         : 1s
  Max response time       : 0s
```

Example: Display detailed group information for specific group and source on selected instance

```
supervisor@rtbrick: op> show igmp group instance default 232.252.0.1 192.0.2.54
(192.0.2.54, 232.252.0.1)
  Outgoing interface      : null0
  Instance                : default
  Source                  : Static
  State                   : No Members Present
  Version                 : IGMP
  Uptime                  : 00h:01m:22s
  Expires                 : n/a
  Membership interval     : n/a
  Last reporter           : n/a
  Last member query count : n/a
  Last member interval    : n/a
  Retransmit time         : n/a
  Max response time       : n/a
(192.0.2.54, 232.252.0.1)
  Outgoing interface      : ppp-0/1/23/72339069014638629
  Instance                : default
  Source                  : Dynamic
  State                   : Members Present
  Version                 : IGMPv3
  Uptime                  : 00h:00m:07s
  Expires                 : 3m 39s
```

```
  Membership interval     : 225s
  Last reporter           : 198.51.100.101
  Last member query count : 3
  Last member interval    : 1s
  Retransmit time         : 1s
  Max response time       : 0s
(192.0.2.54, 232.252.0.1)
  Outgoing interface      : ppp-0/1/23/72339069014638630
  Instance                : default
  Source                  : Dynamic
  State                   : Members Present
  Version                 : IGMPv3
  Uptime                  : 00h:00m:07s
  Expires                 : 3m 39s
  Membership interval     : 225s
  Last reporter           : 198.51.100.102
  Last member query count : 3
  Last member interval    : 1s
  Retransmit time         : 1s
  Max response time       : 0s
(192.0.2.54, 232.252.0.1)
  Outgoing interface      : ppp-0/1/23/72339069014638631
  Instance                : default
  Source                  : Dynamic
  State                   : Members Present
  Version                 : IGMPv3
  Uptime                  : 00h:00m:07s
  Expires                 : 3m 39s
  Membership interval     : 225s
  Last reporter           : 198.51.100.103
  Last member query count : 3
  Last member interval    : 1s
  Retransmit time         : 1s
  Max response time       : 0s
(192.0.2.54, 232.252.0.1)
  Outgoing interface      : ppp-0/1/23/72339069014638632
  Instance                : default
  Source                  : Dynamic
  State                   : Members Present
  Version                 : IGMPv3
  Uptime                  : 00h:00m:07s
  Expires                 : 3m 39s
  Membership interval     : 225s
  Last reporter           : 198.51.100.104
  Last member query count : 3
  Last member interval    : 1s
  Retransmit time         : 1s
  Max response time       : 0s
(192.0.2.54, 232.252.0.1)
  Outgoing interface      : ppp-0/1/23/72339069014638633
  Instance                : default
```

```
  Source                  : Dynamic
  State                   : Members Present
  Version                 : IGMPv3
  Uptime                  : 00h:00m:07s
  Expires                 : 3m 39s
  Membership interval     : 225s
  Last reporter           : 198.51.100.105
  Last member query count : 3
  Last member interval    : 1s
  Retransmit time         : 1s
  Max response time       : 0s
```

Example: Display detailed group information for specific interface.

```
supervisor@rtbrick: op> show igmp group outgoing-interface ppp-
0/1/23/72339069014638629
(192.0.2.54, 232.252.0.1)
  Outgoing interface      : ppp-0/1/23/72339069014638629
  Instance                : default
  Source                  : Dynamic
  State                   : Members Present
  Version                 : IGMPv3
  Uptime                  : 00h:00m:06s
  Expires                 : 3m 40s
  Membership interval     : 225s
  Last reporter           : 198.51.100.101
  Last member query count : 3
  Last member interval    : 1s
  Retransmit time         : 1s
  Max response time       : 0s
```

### IGMP Clear Commands

#### IGMP Interface

**Syntax:**

**clear igmp interface** <attribute> <value>

| Option | Description |
|---|---|
| `<interface_name>` | Clears the specified IGMP interface |
| `instance <instance> statistics` | Clears the IGMP interface statistics for the specified instance. |

Example: Clear interface IGMP statistics

```
supervisor@rtbrick: op> clear igmp interface instance default statistics
```

```
Interface IGMP statistics were successfully cleared
```

**IGMP Group**

**Syntax:**

**clear igmp group** <attribute> <value>

| Option | Description |
|---|---|
| `all` | Clear all IGMP groups present on all instances |
| `instance <instance>` | Clear all IGMP groups present on specific instance |
| `interface <interface_name>` | Clear all IGMP groups present on specific interface |

Example: Clear all IGMP groups present on all instances

```
supervisor@rtbrick>LEAF01: op> clear igmp group all
IGMP groups were successfully cleared
```

# 8.2. PIM

## 8.2.1. PIM Overview

Protocol Independent Multicast (PIM) is a multicast routing protocol that runs over an existing unicast infrastructure. RBFS supports PIM source-specific multicast (SSM). PIM-SSM uses a subset of PIM sparse mode and IGMP version 3 (IGMPv3) to permit a client to receive multicast traffic directly from the source.

PIM SSM builds shortest-path trees (SPTs) rooted at the source immediately because in SSM, the router closest to the interested receiver host is informed of the unicast IP address of the source for the multicast traffic. That is, PIM SSM bypasses the RP connection stage through shared distribution trees, as in PIM sparse mode, and goes directly to the source-based distribution tree.

Internet Protocol Television (IPTV) is a service where digital TV signal data is delivered by using Internet protocol (IP). IPTV service networks typically use PIM-SSM as the multicast routing protocol which has the following characteristics:

- Source specific host membership report for a particular multicast group. IGMPv3 allows a host to describe specific sources from which it would like to receive data.
- PIM shortest path forwarding. Source-specific host report for a particular multicast group and initiating PIM (S,G) joins directly and immediately as result.
- No shared tree forwarding. In order to achieve global effectiveness of SSM, all networks must agree to restrict data forwarding to source trees for some recognized group range. The address range 232.0.0.0/8 has been allocated by IANA for use by source specific multicast (SSM).

#### Supported Platforms

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

### 8.2.2. PIM Configuration

#### Configuration Hierarchy

The diagram illustrates the PIM configuration hierarchy.

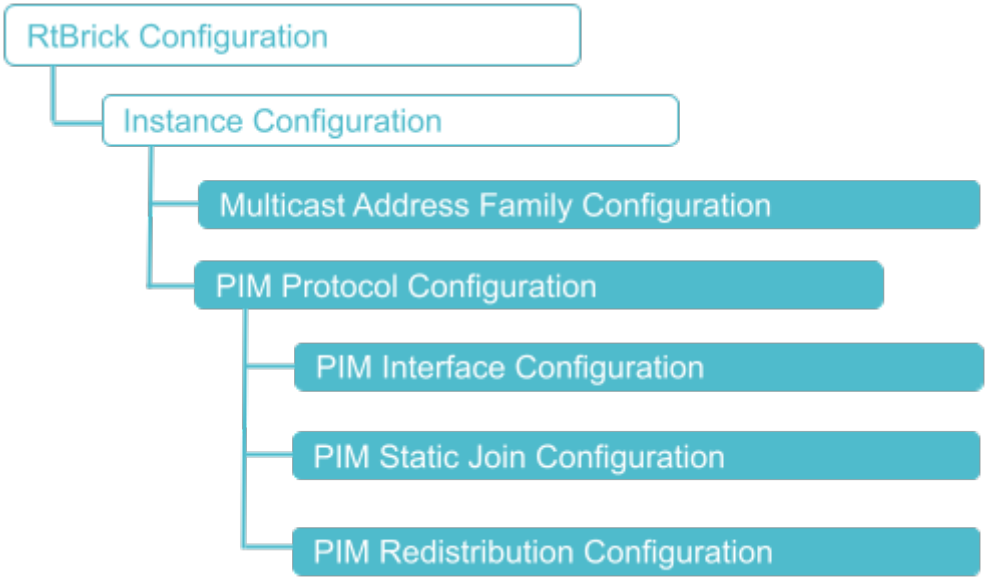



*Figure 148. PIM Configuration Hierarchy*

#### Configuration Syntax and Commands

The following sections describe the interface configuration syntax and commands.

##### Multicast Address Family Configuration

You can enable the multicast IPv4 address family under the PIM instance using the following command:

**Syntax:**

```
set instance <instance> address-family <attribute> <value>
```

| Attribute | Description |
|---|---|
| `<instance>` | Specifies the name of the network instance |
| `<afi>` | Address family identifier (AFI). Supported values: ipv4 |
| `<safi>` | Subsequent address family identifier (SAFI), that is, multicast. |

Example: Multicast Address Family Configuration

```
{
    "rtbrick-config:address-family": [
      {
        "afi": "ipv4",
        "safi": "multicast"
      }
    ]
  }
```

#### PIM Protocol Configuration

**Syntax:**

**set instance** <instance> **protocol pim** <attribute> <value>

If no instance is specified, PIM will be enabled on the default instance. RBFS supports only IPV4 address family.

| Attribute | Description |
|---|---|
| `<instance>` | Specifies the name of the instance |
| `afi <afi>` | Address family identifier (AFI). Supported: ipv4 |
| `join-prune-interval <join-prune-interval>` | PIM join & prune interval. The interval ranges from 10 to 600 seconds. By default, join & prune interval is 60 seconds and hold-down timer is 210 seconds. |
| `sparse-mode interface <…>` | Reference to an entry in the global interface list. Refer to the PIM Interface Configuration section for details on the PIM interface configuration. |
| `sparse-mode static-join <…>` | A static pim join, (*\,G) or (S,G). Refer to the PIM Static Join Configuration section for details on the PIM static join configuration. |
| `sparse-mode redistribute <…>` | Redistribution-related configuration. Refer to the PIM Redistribution Configuration section for details on the PIM redistribution configuration. |

To access the RESTCONF API that corresponds to this CLI, click here.

#### PIM Interface Configuration

**Syntax:**

**set instance** <instance> **protocol pim sparse-mode interface** <attribute> <value>

| Attribute | Description |
|---|---|
| `<instance>` | Specifies the name of the instance |
| `<interface-name>` | Reference to an entry in the global interface list |
| `afi <afi>` | Address family identifier (AFI). Supported: ipv4 |
| `dr-priority <dr-priority>` | Specifies the Specifies the designated router (DR) priority value. |
| `hello-interval <hello-interval>` | Specifies the hello timer in seconds. The hello timer ranges from 1 to 3600 seconds. Default: 30 seconds. |
| `override-interval <override-interval>` | Specifies the override interval in milliseconds. Default: 2000 milliseconds. |
| `propagation-delay <propagation-delay>` | Specifies the propagation delay in milliseconds. Default: 500 milliseconds. |

Example: PIM Interface Configuration

```
{
    "rtbrick-config:interface": [
      {
        "interface-name": "ifl-0/0/0/1",
        "hello-interval": 100,
        "dr-priority": 101
      },
      {
        "interface-name": "ifl-0/0/1/2"
      },
      {
        "interface-name": "ifl-0/0/3/3",
        "propagation-delay": 103,
        "override-interval": 1000
      }
    ]
  }
```

To access the RESTCONF API that corresponds to this CLI, click here.

#### PIM Static Join Configuration

**Syntax:**

**set instance** <instance> **protocol pim sparse-mode static-join** <attribute> <value>

| Attribute | Description |
|---|---|
| `<instance>` | Specifies the name of the instance |
| `<interface-name>` | Reference to an entry in the global interface list |

| Attribute | Description |
|---|---|
| `<outgoing-interface>` | Multicast outbound interface name |
| `<group-address>` | Multicast group IP address |
| `<source-address>` | Multicast source IP address |

Example: PIM Static Join Configuration

```
{
    "rtbrick-config:static-join": [
      {
        "outgoing-interface": "ifl-0/0/0/1",
        "group-address": "198.51.100.12",
        "source-address": "198.51.100.120"
      }
    ]
  }
```

To access the RESTCONF API that corresponds to this CLI, click here.

#### PIM Redistribution Configuration

**Syntax:**

**set instance** <instance> **protocol pim sparse-mode redistribute** <attribute> <value>

**Command Parameters**

| | |
|---|---|
| `<instance>` | Specifies the name of the instance. |
| `<afi>` | Specifies the address family identifier. Supported value: ipv4. |
| `<safi>` | Specifies the subsequent address family identifier. Supported value: multicast. |
| `<source>` | Source protocol from which routes are being redistributed such as BGP or static. |

Example: PIM Redistribution Configuration

```
{
    "rtbrick-config:redistribute": [
      {
        "afi": "ipv4",
        "safi": "multicast",
        "source": "bgp"
      }
    ]
```

```
  }
```

To access the RESTCONF API that corresponds to this CLI, click here.

**PIM Configuration Example**

```
{
    "rtbrick-config:pim": {
      "sparse-mode": {
        "interface": [
          {
            "interface-name": "ifp-0/0/1"
          }
        ],
        "redistribute": [
          {
            "afi": "ipv4",
            "safi": "multicast",
            "source": "bgp"
          }
        ],
        "static-join": [
          {
            "outgoing-interface": "null0",
            "group-address": "198.51.100.110",
            "source-address": "198.18.73.250"
          },
          {
            "outgoing-interface": "null0",
            "group-address": "198.51.100.111",
            "source-address": "198.18.73.250"
          },
          {
            "outgoing-interface": "null0",
            "group-address": "198.51.100.112",
            "source-address": "198.18.73.250"
          },
          {
            "outgoing-interface": "null0",
            "group-address": "198.51.100.113",
            "source-address": "198.18.73.250"
          }
        ]
      }
    }
  }
```

### 8.2.3. PIM Operational Commands

#### PIM Show Commands

##### PIM Interface

**Syntax:**

**show pim interface** <option>

| Option | Description |
|---|---|
| - | Without any option, the commands displays the information for all interfaces. |
| <interface_name> | Displays the PIM interface detail for the default instance |
| instance <instance_name> | Displays interface summary on specific instance |

Example 1: Summary of PIM logical interface for all instances.

```
supervisor@rtbrick: op> show pim interface
Interface             Instance        IP Address        State      DR                Generator ID
ifl-0/0/2/100         ce1-vrf         192.0.2.5         Non-DR     192.0.2.6            846930886
null0                 ce1-vrf         n/a               n/a        n/a                        n/a

ifl-0/0/2/0           default         192.0.2.1         Non-DR     192.0.2.2           1804289383
null0                 default         n/a               n/a        n/a                        n/a
```

Example 2: Summary of interfaces for the specified instance.

```
supervisor@rtbrick: op> show pim interface instance ce1-vrf
Interface             Instance        IP Address        State      DR                Generator ID
ifl-0/0/2/100         ce1-vrf         192.0.2.5         Non-DR     192.0.2.6            846930886
null0                 ce1-vrf         n/a               n/a        n/a                        n/a
```

Example 3: Detailed view of the specified PIM interface.

```
supervisor@rtbrick: op> show pim interface ifl-0/0/2/100
Interface: ifl-0/0/2/100
  Instance                 : ce1-vrf
  State                    : Non-DR
  Primary address          : 192.0.2.5
  Generation ID            : 846930886
  Timer values
    Hello interval         : 30s
    Join/Prune interval    : 60s
    Hold interval          : 90s
    Override interval      : 2000ms
```

```
    Prune delay interval   : 500ms
  DR election
    DR address             : 192.0.2.6
    DR priority            : 1
    DR election count      : 2
  Negotiated
    DR priority used       : True
    Lan delay used         : False
    Lan prune interval     : 0
    Lan override  used     : False
    Lan override interval  : 0
  Statistics
    Hello
      Received             : 1
      Sent                 : 1
    Membership
      Received             : 0
      Sent                 : 0
    Assert
      Received             : 0
      Sent                 : 0
    JP Timers
      Early Termination    : 0
```

API To access the Operational State API that corresponds to this CLI, click include::common:partial$operational_api_links.adoc[tag=show_pim_interface].

#### PIM Membership

**Syntax:**

**show pim membership** <option>

| Option | Description |
|---|---|
| - | Without any option, the commands displays PIM membership summary on all instances. |
| instance <instance_name> | Displays the PIM membership summary information on specific instance. |
| detail | Detailed information on all BGP peers in all instances in a list view. |
| instance <instance_name> detail | Displays the PIM membership detailed information on specific instance. |
| <group_address> | Specifies the multicast multicast group address |
| <source_address> | Specifies the source from which the multicast traffic is received |

Example 1: Summary of the PIM membership on all instances.

```
supervisor@rtbrick: op> show pim membership
Source               Group                Interface                     Instance
192.0.2.60           232.252.0.11         null0                         ce1-vrf
192.0.2.60           232.252.0.12         null0                         ce1-vrf
192.0.2.60           232.252.0.13         null0                         ce1-vrf
192.0.2.54           232.252.0.1          null0                         default
192.0.2.54           232.252.0.2          null0                         default
192.0.2.54           232.252.0.3          null0                         default
```

Example 2: Summary of the PIM membership for the specified instance.

```
supervisor@rtbrick: op> show pim membership instance default
Source               Group                Interface                     Instance
192.0.2.54           232.252.0.1          null0                         default
192.0.2.54           232.252.0.2          null0                         default
192.0.2.54           232.252.0.3          null0                         default
```

Example 3: Detailed view of PIM membership on all instances.

```
supervisor@rtbrick: op> show pim membership detail
192.0.2.60, 232.252.0.11
  Instance           : ce1-vrf
  Outgoing interface : null0
  Source             : pim
  Subtype            : Join
  Subsource          : Static
192.0.2.60, 232.252.0.12
  Instance           : ce1-vrf
  Outgoing interface : null0
  Source             : pim
  Subtype            : Join
  Subsource          : Static
192.0.2.60, 232.252.0.13
  Instance           : ce1-vrf
  Outgoing interface : null0
  Source             : pim
  Subtype            : Join
  Subsource          : Static
192.0.2.54, 232.252.0.1
  Instance           : default
  Outgoing interface : null0
  Source             : pim
  Subtype            : Join
  Subsource          : Static
192.0.2.54, 232.252.0.2
  Instance           : default
  Outgoing interface : null0
  Source             : pim
```

```
  Subtype            : Join
  Subsource          : Static
192.0.2.54, 232.252.0.3
  Instance           : default
  Outgoing interface : null0
  Source             : pim
  Subtype            : Join
  Subsource          : Static
```

Example 4: Detailed view of PIM membership for the specified instance.

```
supervisor@rtbrick: op> show pim membership instance default detail
192.0.2.54, 232.252.0.1
  Instance           : default
  Outgoing interface : null0
  Source             : pim
  Subtype            : Join
  Subsource          : Static
192.0.2.54, 232.252.0.2
  Instance           : default
  Outgoing interface : null0
  Source             : pim
  Subtype            : Join
  Subsource          : Static
192.0.2.54, 232.252.0.3
  Instance           : default
  Outgoing interface : null0
  Source             : pim
  Subtype            : Join
  Subsource          : Static
```

Example 5: Detailed view of PIM membership for the specified group-address and source-address in selected instance.

```
supervisor@rtbrick: op> show pim membership instance default 232.252.0.1 192.0.2.54
192.0.2.54, 232.252.0.1
  Instance           : default
  Outgoing interface : null0
  Source             : pim
  Subtype            : Join
  Subsource          : Static
```

#### PIM Join and Prune

**Syntax:**

**show pim join-prune** <option>

| Option | Description |
|---|---|
| - | Without any option, the commands displays join and prune summary command on all instances. |
| instance <instance_name> | Displays join and prune summary on a specific instance. |
| detail | Displays PIM join and prune detailed information on all instances. |
| instance <instance_name> detail | Displays detailed join and prune information in selected instance. |
| <group_address> | Displays PIM join and prune detailed information for a specific group on all instance |
| <source_address> | Specifies the source from which the multicast traffic is received. |

Example 1: Summary of the PIM join and prune all instances.

```
supervisor@rtbrick: op> show pim join-prune
Source                Group                 Upstream Interface    Instance
192.0.2.60            232.252.0.11          ifl-0/0/2/100         ce1-vrf
192.0.2.60            232.252.0.12          ifl-0/0/2/100         ce1-vrf
192.0.2.60            232.252.0.13          ifl-0/0/2/100         ce1-vrf
192.0.2.54            232.252.0.1           ifl-0/0/2/0           default
192.0.2.54            232.252.0.2           ifl-0/0/2/0           default
192.0.2.54            232.252.0.3           ifl-0/0/2/0           default
```

Example 2: Summary of the PIM join and prune summary for the specified instance.

```
supervisor@rtbrick: op> show pim join-prune instance ce1-vrf
Source                Group                 Upstream Interface    Instance
192.0.2.60            232.252.0.11          ifl-0/0/2/100         ce1-vrf
192.0.2.60            232.252.0.12          ifl-0/0/2/100         ce1-vrf
192.0.2.60            232.252.0.13          ifl-0/0/2/100         ce1-vrf
```

Example 3: Detailed view of the PIM join and prune for the specified instance.

```
supervisor@rtbrick: op> show pim join-prune instance ce1-vrf detail
192.0.2.60, 232.252.0.11
  Instance           : ce1-vrf
  Upstream interface : ifl-0/0/2/100
  Upstream neighbor  : 192.0.2.6
  Type               : Join
192.0.2.60, 232.252.0.12
  Instance           : ce1-vrf
```

```
  Upstream interface : ifl-0/0/2/100
  Upstream neighbor  : 192.0.2.6
  Type               : Join
192.0.2.60, 232.252.0.13
  Instance           : ce1-vrf
  Upstream interface : ifl-0/0/2/100
  Upstream neighbor  : 192.0.2.6
  Type               : Join
```

Example 4: Detailed view of the PIM join and prune for the specified instance, source, and group.

```
supervisor@rtbrick: op> show pim join-prune instance ce1-vrf 232.252.0.11 192.0.2.60
192.0.2.60, 232.252.0.11
  Instance           : ce1-vrf
  Upstream interface : ifl-0/0/2/100
  Upstream neighbor  : 192.0.2.6
  Type               : Join
```

#### PIM Neighbors

**Syntax:**

**show pim neighbor** <option>

| Option | Description |
|---|---|
| - | Without any option, the commands displays the PIM neighbor summary on all instances. |
| instance <instance_name> | Displays the PIM neighbor summary information on specific instance |
| instance <instance_name> <neighbor_address> | Displays detailed information for specific PIM neighbor in selected instance. |

Example 1: Summary of PIM neighbor on all instances.

```
supervisor@rtbrick: op> show pim neighbor
Neighbor         Interface           Instance       State          Generation ID  Uptime         Expires
192.0.2.6        ifl-0/0/2/100       ce1-vrf        Up                 846930886  00h:00m:18s    1m 27s
192.0.2.2        ifl-0/0/2/0         default        Up                1804289383  00h:00m:18s    1m 27s
```

Example 2: Summary of PIM neighbor for the specified instance.

```
supervisor@rtbrick: op> show pim neighbor instance default
Neighbor         Interface           Instance       State          Generation ID  Uptime         Expires
192.0.2.2        ifl-0/0/2/0         default        Up                1804289383  00h:00m:19s    1m 26s
```

Example 3: Detailed view of PIM neighbor for the specified instance.

```
supervisor@rtbrick: op> show pim neighbor instance default 192.0.2.2
Neighbor: 192.0.2.2
  Interface            : ifl-0/0/2/0
  Instance             : default
  Hold down interval   : 105s
  Expires              : 1m 24ss
  Generation ID        : 1804289383
  DR priority          : 1
  Uptime               : 00h:00m:20s
  Last transition time : Fri Jun 05 14:52:05 GMT +0000 2026
  Holddown received    : 1
```

API To access the Operational State API that corresponds to this CLI, click include::common:partial$operational_api_links.adoc[tag=show_pim_neighbor].

#### PIM Reverse Path Forwarding (RPF)

**Syntax:**

**show pim rpf** <option>

| Option | Description |
|---|---|
| - | Without any option, the commands displays the PIM rpf summary information on all instance. |
| instance <instance_name> | Displays the PIM rpf summary information on specific instance. |
| instance <instance> <source_address> | Displays the PIM rpf detailed information for specific source-address in selected instance. |

Example 1: Summary of PIM rpf on all instances.

```
supervisor@rtbrick: op> show pim rpf
Multicast Source  Instance       RPF Interface       Neighbor
192.0.2.60        ce1-vrf        ifl-0/0/2/100       n/a
192.0.2.54        default        ifl-0/0/2/0         n/a
```

Example 2: Summary of PIM rpf for the specified instance.

```
supervisor@rtbrick: op> show pim rpf instance ce1-vrf
Multicast Source  Instance       RPF Interface       Neighbor
192.0.2.60        ce1-vrf        ifl-0/0/2/100       n/a
```

Example 3: Detailed view of PIM rpf for the specified source-address in selected instance.

```
supervisor@rtbrick: op> show pim rpf instance ce1-vrf 192.0.2.60
Multicast source : 192.0.2.60
  Instance        : ce1-vrf
  AFI             : ipv4
  SAFI            : unicast
  RPF interface   : ifl-0/0/2/100
  Peer            : n/a
  Covering prefix : 192.0.2.60/32
  MAC address     : n/a
```

#### PIM Routes

**Syntax:**

**show pim mroute** <option>

| Option | Description |
|---|---|
| - | Without any option, the commands displays the PIM routes summary information on all instance. |
| `detail` | Displays the PIM routes detail for all instances. |
| `instance <instance_name>` | Displays the PIM routes summary on specific instance |
| `instance <instance_name> detail` | Displays the PIM routes detailed information on specific instances. |

Example 1: Summary of PIM routes for all instances.

```
supervisor@rtbrick: op> show pim mroute
Instance: ce1-vrf, AFI: ipv4, SAFI: multicast
  Source              Group               Route Source   Preference  Nexthop             OIF
  192.0.2.60          232.252.0.11        pim            240         n/a                 null0
  192.0.2.60          232.252.0.12        pim            240         n/a                 null0
  192.0.2.60          232.252.0.13        pim            240         n/a                 null0
Instance: default, AFI: ipv4, SAFI: multicast
  Source              Group               Route Source   Preference  Nexthop             OIF
  192.0.2.54          232.252.0.1         pim            240         n/a                 null0
  192.0.2.54          232.252.0.2         pim            240         n/a                 null0
  192.0.2.54          232.252.0.3         pim            240         n/a                 null0
```

Example 2: Detailed view of PIM routes for all instances.

```
supervisor@rtbrick: op> show pim mroute detail
192.0.2.60, 232.252.0.11
  Source       : pim                    Preference       : 240
  Sub source   : Static                 Subtype          : Join
  RPF neighbor : 192.0.2.6              RPF interface    : ifl-0/0/2/100
```

```
  Nexthop      : n/a                    Egress interface : null0
  Nexthop type : Multicast Fanout       NextHop action   : None
  Destination  : ce1-vrf-ipv4-multicast
  Resolved in  : ce1-vrf-ipv4-multicast
192.0.2.60, 232.252.0.12
  Source       : pim                    Preference       : 240
  Sub source   : Static                 Subtype          : Join
  RPF neighbor : 192.0.2.6              RPF interface    : ifl-0/0/2/100
  Nexthop      : n/a                    Egress interface : null0
  Nexthop type : Multicast Fanout       NextHop action   : None
  Destination  : ce1-vrf-ipv4-multicast
  Resolved in  : ce1-vrf-ipv4-multicast
192.0.2.60, 232.252.0.13
  Source       : pim                    Preference       : 240
  Sub source   : Static                 Subtype          : Join
  RPF neighbor : 192.0.2.6              RPF interface    : ifl-0/0/2/100
  Nexthop      : n/a                    Egress interface : null0
  Nexthop type : Multicast Fanout       NextHop action   : None
  Destination  : ce1-vrf-ipv4-multicast
  Resolved in  : ce1-vrf-ipv4-multicast
192.0.2.54, 232.252.0.1
  Source       : pim                    Preference       : 240
  Sub source   : Static                 Subtype          : Join
  RPF neighbor : 192.0.2.2              RPF interface    : ifl-0/0/2/0
  Nexthop      : n/a                    Egress interface : null0
  Nexthop type : Multicast Fanout       NextHop action   : None
  Destination  : default-ipv4-multicast
  Resolved in  : default-ipv4-multicast
192.0.2.54, 232.252.0.2
  Source       : pim                    Preference       : 240
  Sub source   : Static                 Subtype          : Join
  RPF neighbor : 192.0.2.2              RPF interface    : ifl-0/0/2/0
  Nexthop      : n/a                    Egress interface : null0
  Nexthop type : Multicast Fanout       NextHop action   : None
  Destination  : default-ipv4-multicast
  Resolved in  : default-ipv4-multicast
192.0.2.54, 232.252.0.3
  Source       : pim                    Preference       : 240
  Sub source   : Static                 Subtype          : Join
  RPF neighbor : 192.0.2.2              RPF interface    : ifl-0/0/2/0
  Nexthop      : n/a                    Egress interface : null0
  Nexthop type : Multicast Fanout       NextHop action   : None
  Destination  : default-ipv4-multicast
  Resolved in  : default-ipv4-multicast
```

Example 3: Summary of PIM routes for the specific instance.

```
supervisor@rtbrick: op> show pim mroute instance default
Instance: default, AFI: ipv4, SAFI: multicast
  Source              Group               Route Source   Preference  Nexthop             OIF
```

```
  192.0.2.54          232.252.0.1         pim            240         n/a                null0
  192.0.2.54          232.252.0.2         pim            240         n/a                null0
  192.0.2.54          232.252.0.3         pim            240         n/a                null0
```

Example 4: Detailed view of PIM routes for the specified instances.

```
supervisor@rtbrick: op> show pim mroute instance default detail
192.0.2.54, 232.252.0.1
  Source       : pim                    Preference       : 240
  Sub source   : Static                 Subtype          : Join
  RPF neighbor : 192.0.2.2              RPF interface    : ifl-0/0/2/0
  Nexthop      : n/a                    Egress interface : null0
  Nexthop type : Multicast Fanout       NextHop action   : None
  Destination  : default-ipv4-multicast
  Resolved in  : default-ipv4-multicast
192.0.2.54, 232.252.0.2
  Source       : pim                    Preference       : 240
  Sub source   : Static                 Subtype          : Join
  RPF neighbor : 192.0.2.2              RPF interface    : ifl-0/0/2/0
  Nexthop      : n/a                    Egress interface : null0
  Nexthop type : Multicast Fanout       NextHop action   : None
  Destination  : default-ipv4-multicast
  Resolved in  : default-ipv4-multicast
192.0.2.54, 232.252.0.3
  Source       : pim                    Preference       : 240
  Sub source   : Static                 Subtype          : Join
  RPF neighbor : 192.0.2.2              RPF interface    : ifl-0/0/2/0
  Nexthop      : n/a                    Egress interface : null0
  Nexthop type : Multicast Fanout       NextHop action   : None
  Destination  : default-ipv4-multicast
  Resolved in  : default-ipv4-multicast
```

API To access the Operational State API that corresponds to this CLI, click include::common:partial$operational_api_links.adoc[tag=show_pim_mroute].

### PIM Clear Commands

Clear commands allow to reset operational states.

#### PIM Neighbor

**Syntax:**

```
clear pim neighbor
```

| Option | Description |
|---|---|
| `neighbor` | Clears all neighbors. |

Example: The example below shows how to clear pim neighbor.

```
supervisor@rtbrick: op> clear pim neighbor
All instance clear triggered
```

**PIM Neighbor Instance**

**Syntax:**

**clear pim neighbor instance** <option> ...

| Option | Description |
|---|---|
| `neighbor instance <instance_name>` | Clears all neighbors on the specified instance. |
| `neighbor instance <instance_name> interface <interface-name>` | Clears all neighbors on the specified interface. |

Example 1: The example below shows how to clear neighbors of the specified instance

```
supervisor@rtbrick: op> clear pim neighbor instance vrf1
Instance clear triggered
```

Example 2: The example below shows how to clear neighbors of the specified interface

```
supervisor@rtbrick: op> clear pim neighbor instance vrf1 interface ifl-0/0/0
Instance clear triggered
```

# 8.3. Multicast VPN

## 8.3.1. MVPN Overview

The Multicast VPN (MVPN) feature provides the ability to support multicast over a Layer 3 VPN. Multicast allows the efficient distribution of information between a single multicast source and multiple receivers. IP multicast is used to stream video, voice, and data to an MPLS VPN network core. The RBFS MVPN implementation is based on RFC 6513 “Multicast in MPLS/BGP IP VPNs” and RFC 6514 “BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs”.

RFC and draft compliance are partial except as specified.

RBFS can operate in a spine/leaf fabric as shown in the diagram. The leaf device delivers access services to subscribers or assets, and the spine device provides connectivity to the core network. In such a scenario, IPv4 multicast traffic is carried in a multicast VPN instance. This also allows to deliver IPv4 multicast traffic across an IPv6-only fabric. In terms of MVPN, both leaf and spine

devices act as Provider Edge (PE) routers.

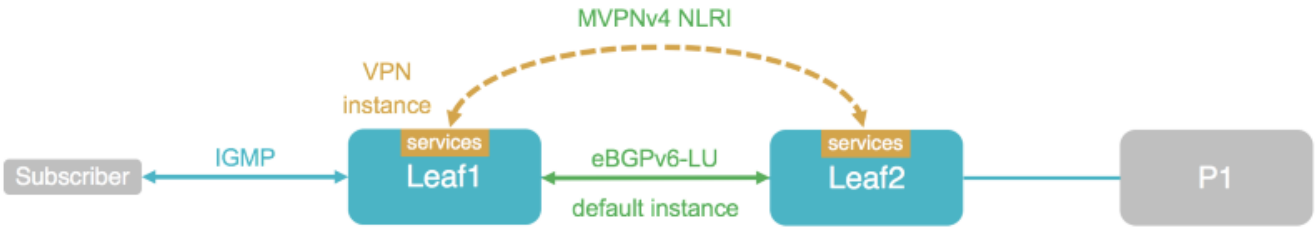



*Figure 149. Multicast VPN Scenario*

Multicast subscribers will typically join multicast streams via IGMP. Leaf switches will translate and signal the join messages to the spines using BGP MVPN routes. Spine switches will further forward the join messages towards the source to the upstream core router.

#### Multicast Address Families

In MVPN, there are two types of instances that address families involved:

- Within the multicast VPN instance, joins are represented as multicast routes. These are AFI 1 (IPv4), SAFI 2 (Multicast). The IPv4 multicast address family needs to be enabled in the VPN service instance as well as for BGP in the VPN instance.
- In the Multicast VPN itself, messages like joins or active sources are advertised using BGP MVPN routes. These are AFI 1 (IPv4), SAFI 5 (MVPN). The BGP peerings that carry the MVPN routes typically run in the default instance. Therefore the MVPN address family needs to be enabled in the default instance, both for the BGP instance itself as well as for the BGP peerings.

In summary, the following address families need to be enabled by configuration:

- The IPv4 multicast address family in the VPN service instance.
- The IPv4 multicast address family for BGP in the VPN service instance.
- The Multicast VPN address family in the BGP default instance.
- The Multicast VPN address family in the BGP peerings in the default instance.

#### Supported Platforms

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

### 8.3.2. MVPN Configuration

#### Configuration Hierarchy

The diagram illustrates the Multicast VPN configuration hierarchy.

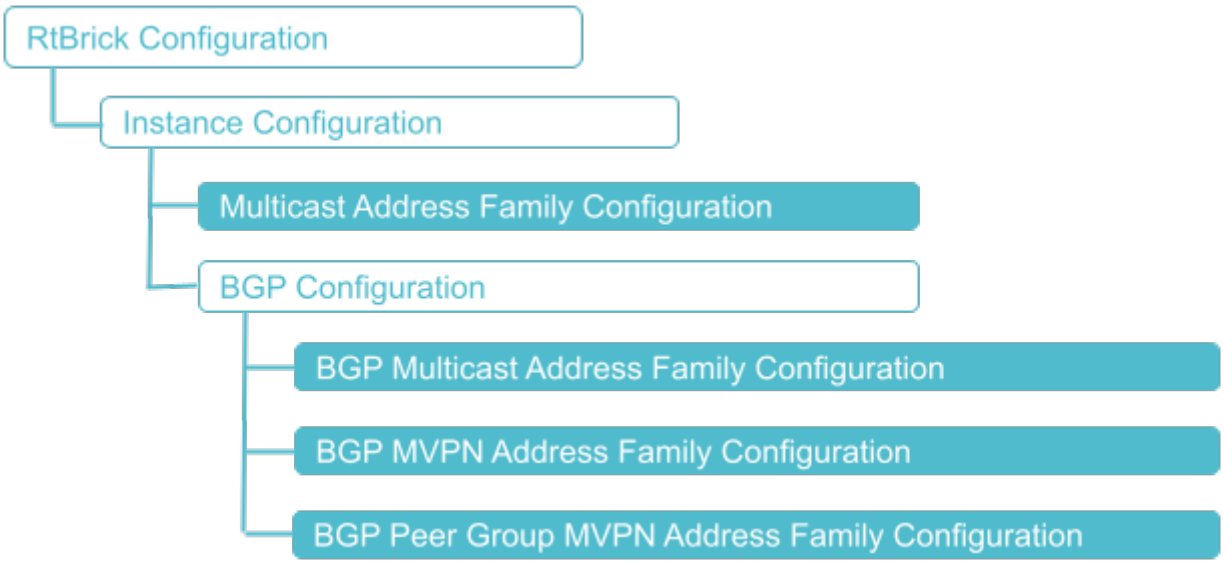



*Figure 150. Multicast VPN Configuration Hierarchy*

### Configuration Syntax and Commands

The following sections describe the MVPN configuration syntax and commands.

#### Multicast Address Family Configuration

Enable the IPv4 multicast address family in the VPN service instance.

**Syntax:**

**set instance** <instance> **address-family** <afi> <safi> <attribute> <value>

| Attribute | Description |
|---|---|
| `<instance>` | The name of the VPN service instance |
| `<afi>` | Address family identifier (AFI). For IPv4 multicast, the required value is: `ipv4` |
| `<safi>` | Subsequent address family identifier (SAFI). For IPv4 multicast, the required value is: `multicast` |

Example: Instance Address Family Configuration

```
{
  "ietf-restconf:data": {
    "rtbrick-config:instance": [
      {
        "name": "services",
        "address-family": [
          {
            "afi": "ipv4",
            "safi": "multicast",
            "route-target": {
```

```
              "import": "target:198.51.100.10:13",
              "export": "target:198.51.100.10:13"
            }
          }
        ]
      }
    ]
  }
}
```

#### BGP Multicast Address Family Configuration

Enable the IPv4 multicast address family for BGP in the VPN service instance.

**Syntax:**

**set instance** <instance> protocol bgp **address-family** <afi> <safi> <attribute> <value>

| Attribute | Description |
|---|---|
| `<instance>` | The name of the VPN service instance |
| `<afi>` | Address family identifier (AFI). For IPv4 multicast, the required value is: `ipv4` |
| `<safi>` | Subsequent address family identifier (SAFI). For IPv4 multicast, the required value is: `multicast` |
| `redistribute <source>` | Optionally redistribute PIM routes into BGP to translate and advertise subscriber join message as MVPN routes. The required source value is `pim` as IGMP routes are handled by the PIM process too. |

Example: Instance BGP Address Family Configuration

```
{
  "ietf-restconf:data": {
  "rtbrick-config:instance": {
    "name": "services",
    "protocol": {
      "bgp": {
        "address-family": [
          {
            "afi": "ipv4",
            "safi": "multicast",
            "redistribute": [
              {
                "source": "pim"
              }
            ]
```

```
          }
        ]
      }
    }
  }
}
```

#### BGP MVPN Address Family Configuration

Enable the MVPN address family in the global BGP instance.

**Syntax:**

**set instance** <instance> **protocol bgp address-family** <afi> <safi> <attribute> <value>

| Attribute | Description |
|---|---|
| `<instance>` | Name of the global BGP instance. Typically this will be the `default` instance. |
| `<afi>` | Address family identifier (AFI). For MVPN, the required value is: `ipv4` |
| `<safi>` | Supported SAFIs are unicast. For MVPN, the required value is: `vpn-multicast` |

Example: BGP MVPN Address Family Configuration

```
{
  "ietf-restconf:data": {
    "rtbrick-config:instance": [
      {
        "name": "default",
        "protocol": {
          "bgp": {
            "address-family": [
              {
                "afi": "ipv4",
                "safi": "vpn-multicast"
              }
            ]
          }
        }
      }
    ]
  }
}
```

#### BGP Peer Group MVPN Address Family Configuration

Enable the MVPN address family for the global BGP peerings carrying the MVPN routes.

**Syntax:**

**set instance** <instance> **protocol bgp peer-group** <attribute> <value>

| Attribute | Description |
|---|---|
| `<instance>` | Name of the global BGP instance. Typically this will be the `default` instance. |
| `<pg-name>` | Peer group name |
| `address-family <afi> <safi>` | BGP peer group address family specific. For MVPN, the required AFI/SAFI values are: `ipv4`/ `vpn-multicast`. |

Example: BGP Peer Group MVPN Address Family Configuration

```
{
  "ietf-restconf:data": {
    "rtbrick-config:peer-group": [
      {
        "pg-name": "spine",
        "address-family": [
          {
            "afi": "ipv4",
            "safi": "vpn-multicast",
            "extended-nexthop": "true",
            "update-nexthop": {
              "ipv6-address": "2001:db8:0:103::"
            }
          }
        ]
      }
    ]
  }
```

#### Multicast VPN Configuration Example

```
{
  "ietf-restconf:data": {
    "rtbrick-config:instance": [
      {
        "name": "default",
        "protocol": {
          "bgp": {
            "address-family": [
```

```
            {
              "afi": "ipv4",
              "safi": "vpn-multicast"
            }
          ],
          "peer-group": [
            {
              "pg-name": "spine",
              "address-family": [
                {
                  "afi": "ipv4",
                  "safi": "vpn-multicast",
                  "extended-nexthop": "true",
                  "update-nexthop": {
                    "ipv6-address": "2001:db8:0:103::"
                  }
                }
              ]
            }
          ]
        }
      }
    },
    {
      "name": "services",
      "address-family": [
        {
          "afi": "ipv4",
          "safi": "multicast",
          "route-target": {
            "import": "target:198.51.100.10:13",
            "export": "target:198.51.100.10:13"
          }
        }
      ],
      "protocol": {
        "bgp": {
          "address-family": [
            {
              "afi": "ipv4",
              "safi": "multicast",
              "redistribute": [
                {
                  "source": "pim"
                }
              ]
            }
          ]
        }
      }
    }
```

```
    ]
  }
}
```

### 8.3.3. MVPN Operational Commands

**Show Commands**

**Syntax:**

**show mroute** <options>

| Option | Description |
|---|---|
| - | Without any option, the commands displays summary of all the multicast routes. |
| all | Displays multicast routes for all the VPN instances. |
| detail | Displays detailed view of all the multicast routes including the statistics and ingress interface details. |
| detail all | Displays detailed view of all the multicast routes for all the VPN instances. |
| instance <vpn-instance-name> | Name of the VPN instance |
| group <group-address> | Multicast group address |
| <ipv4\|ipv6> instance <instance_name> source <source-address> | Multicast source address |

The following example output shows the mroute details for two PPPoE subscribers joining two different multicast group, where it displays the multicast traffic statistics along with nexthop information, ingress and egress interface.

Multicast traffic statistics is fetched at an interval of 60 seconds

```
supervisor@rtbrick: op> show mroute detail
Instance: ce2-vrf, AFI: ipv4, SAFI: multicast
192.0.2.54/32, 232.252.0.1/32
  Source: bgp, Preference: 20
  Forwarded packets: 0, Forwarded bytes: 0, Forwarded pps: 0
  Dropped packets: 0, Dropped bytes: 0, Dropped pps: 0
      Next Hop type: Multicast Fanout, Next Hop action: MPLS Replicate
      Resolved in: default-ipv4-labeled-unicast
      Egress interface: ifl-0/0/7/0, NextHop MAC: 98:19:2c:06:b5:03
      MPLS-Label: 519938,bos:1
192.0.2.54/32, 232.252.0.2/32
  Source: bgp, Preference: 20
```

```
  Forwarded packets: 0, Forwarded bytes: 0, Forwarded pps: 0
  Dropped packets: 0, Dropped bytes: 0, Dropped pps: 0
      Next Hop type: Multicast Fanout, Next Hop action: MPLS Replicate
      Resolved in: default-ipv4-labeled-unicast
      Egress interface: ifl-0/0/7/0, NextHop MAC: 98:19:2c:06:b5:03
      MPLS-Label: 519938,bos:1
```

### MVPN Route Show Commands

**Syntax:**

**show bgp fib ipv4 vpn-multicast instance default** <attribute> <value>

| Option | Description |
|---|---|
| - | Without any option, the commands displays summary of all the multicast routes. |
| detail | Displays detailed view of all the VPN multicast routes. |
| <prefix> | Destination prefix address |

Example: Display MVPN Route Summary Information

```
supervisor@rtbrick: op> show bgp fib ipv4 vpn-multicast instance default
Instance: default, AFI: ipv4, SAFI: vpn-multicast, Total routes: 4
  Group           Source                                      Preference      Out Label            Route Type
Next Hop
  -               -                                           20              519938,bos:1         Intra-AS_I-PMSI_AD
198.51.100.11
  -               -                                           200             519938,bos:1         Intra-AS_I-PMSI_AD
-
  232.252.0.1/32  192.0.2.54/32                               20              519938,bos:1         Source_Tree_Join
198.51.100.11
  232.252.0.2/32  192.0.2.54/32                               20              519938,bos:1         Source_Tree_Join
198.51.100.11
```

### Clear Commands

#### Clearing Multicast Statistics

Use the `clear multicast statistics` command to reset multicast traffic counters to zero. After the command completes successfully, the counters begin collecting new statistics.

**Syntax:**

**clear multicast statistics**

The example below shows the command to clear multicast statistics.

```
supervisor@rtbrick.net: cfg> clear multicast statistics
Success : command success
```

# 9. Security

The Security section of the RtBrick FullStack (RBFS) User Guides covers essential mechanisms for securing both the management and control planes of the network.

Key Features:

- Explains how to harden and protect both the network management and control planes from unauthorized access.
- Provides commands to manage local users, credentials, and administrative permissions.
- Details how to use BGP FlowSpec to dynamically block DDoS attacks and filter malicious traffic.
- Covers BGP RPKI to prevent route hijacking and sFlow to monitor real-time network traffic patterns

The following guides are available under the Security section:

- Securing Management Plane
- Securing Control Plane
- Local User Management
- BGP FlowSpec
- BGP RPKI
- sFlow

## 9.1. Securing Management Plane

### 9.1.1. Securing the Management Plane Overview

The Securing Management Plane feature provides the capability to restrict the access to the management plane only to authenticated and authorized subjects.

The authentication identifies a subject, and the authorization validates if the subject is allowed to execute the action.

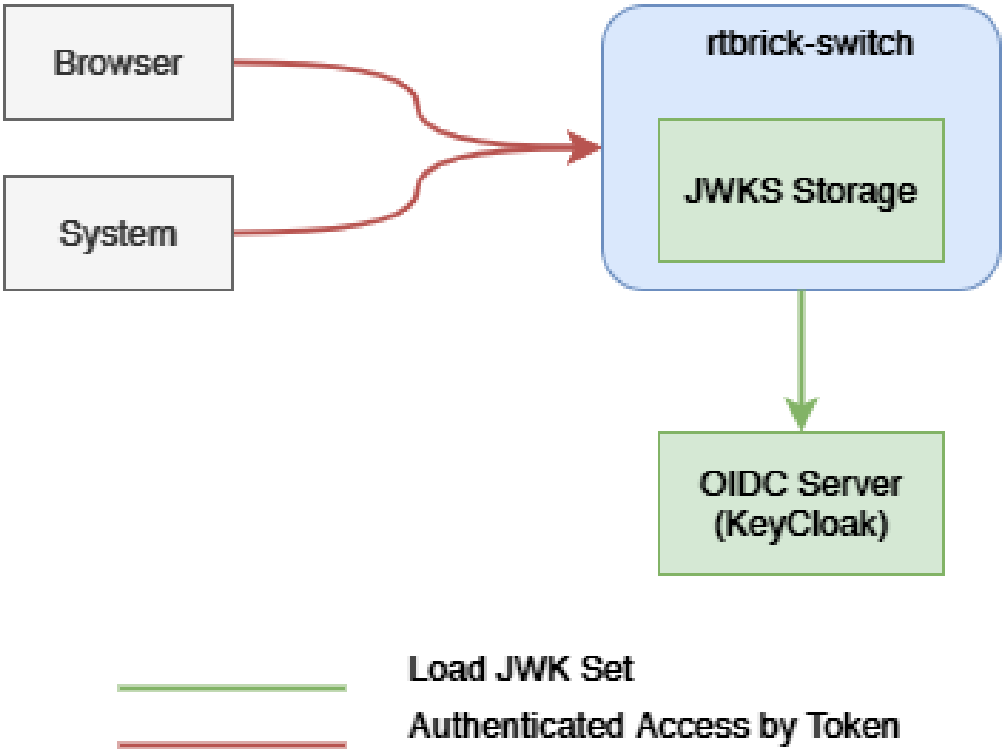



*Figure 151. External Dataflow*

The figure-1 shows the data flow when accessing an rtbrick-switch. Each call against the switch in more detail against the API Gateway Daemon (APIGWD) of the switch has to be authenticated with an access token. There is only one exception when accessing the CTLRD's UI; it is possible to be redirected to an OpenIDConnect Authenticator.

The APIGWD validates the access token against an JSON Web Key Set (JWKS) (https://tools.ietf.org/html/rfc7517). This key set can be loaded from a file locally on the system or auto discovered via the OpenIDConnect server.

A valid access token, in the sense of syntactically correct but also successfully validated signature by one of the JSON Web Key of the JWKS files, leads in an authenticated user. If the validation is unsuccessful, the call will be rejected.

The access token contains scopes which are used internally for the authorization checks. The authorization is a role based authorization where the scopes equal to the roles.

Internally the access token is converted to an RtBrick token, and all the communications inside the switch is authenticated via this RtBrick token.

The dataflow inside of the switch can be seen in Figure 2.

The scopes of the access token are copied to the RtBrick Token.

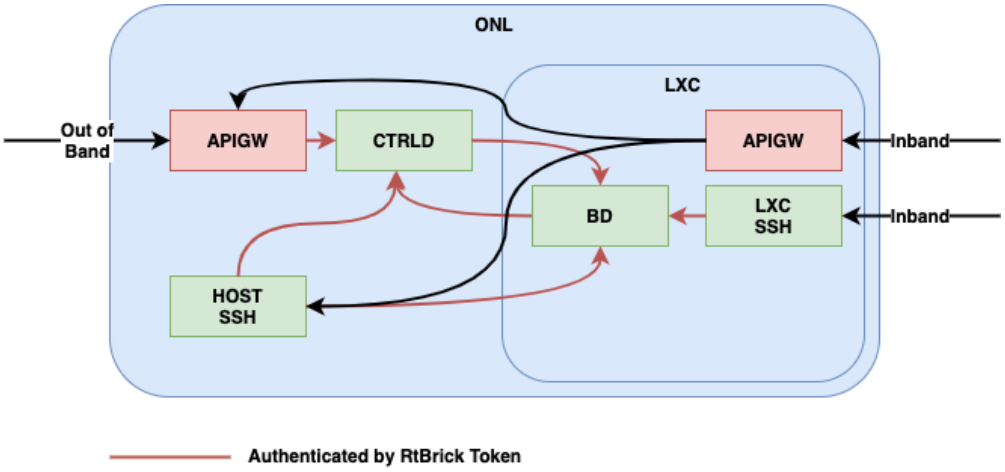



*Figure 152. Internal Dataflow*

#### Supported Platforms

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

### 9.1.2. RtBrick Token

#### JSON Web Tokens

JSON Web Token (JWT) is an open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object.

This information can be verified and trusted because it is digitally signed. JWTs can be signed using a secret (with the HMAC algorithm) or a public/private key pair using RSA or ECDSA.

RFC and draft compliance are partial except as specified.

For more information about JSON Web Token, see https://jwt.io/introduction/.

##### Structure

In its compact form, JSON Web Tokens consist of three parts separated by dots (.), which are:

- Header
- Payload
- Signature

Therefore, a JWT typically looks like the following.

```
xxxxx.yyyyy.zzzzz
```

##### Header

The header typically consists of two parts:

- The **type of the token**, which is JWT
- The signing algorithm that is being used, such as HMAC SHA256 or RSA

The suite of specifications on JWT provisions a few different options to identify particular cryptographic keys. The most straightforward mechanism is the "kid" claim. This claim can be added to the header of the token. It is intended to contain a string-based key identifier.

For example:

```
{
  "alg": "HS256",
  "typ": "JWT",
  "kid": "0815"
}
```

Then, this JSON is Base64Url encoded to form the first part of the JWT.

#### Payload

The second part of the token is the **payload**, which contains the claims. Claims are statements about an entity (typically, the user) and additional data.

There are three types of claims:

- registered
- public
- private

**Registered claims**

These are a set of predefined claims which are not mandatory but recommended, to provide a set of useful, interoperable claims. Some of them are: iss (issuer), exp (expiration time), sub (subject), aud (audience), and others.

**Public claims**

These can be defined at will by those using JWTs. But to avoid collisions they should be defined in the IANA JSON Web Token Registry or be defined as a URI that contains a collision resistant namespace.

**Private claims**

These are the custom claims created to share information between parties that agree on using them and are neither registered or public claims.

An example payload is as follows:

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "iat": 1516239022,
```

```
  "exp": 1600000000,
  "scope": "user"
}
```

The payload is then Base64Url encoded to form the second part of the JSON Web Token.

#### Signature

To create the signature part you have to take the encoded header, the encoded payload, a secret, the algorithm specified in the header, and sign that.

For example if you want to use the HMAC SHA256 algorithm, the signature will be created in the following way:

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  secret)
```

The signature is used to verify the message wasn't changed along the way, and, in the case of tokens signed with a private key, it can also verify that the sender of the JWT is who it says it is.

#### Putting all together

The output is three Base64-URL strings separated by dots that can be easily passed in HTML and HTTP environments.

#### AccessToken

The Access Token is a JSON Web Token. The token is typically sent in the Authorization header using the Bearer schema. The content of the header should look like the following:

```
Authorization: Bearer <token>
```

The API Gateway also supports sending the token as Cookie, but this is not described here, that is only used for the CTRLD web UI.

The token has to have the `kid` claim in the header. This kid is used to find the right JSON Web Key (JWK) from one of the JSON Web Key Sets (JWKS).

APIGWD searches for the JWKS file under `/etc/rtbrick/apigwd/access_secret_jwks.json` or if `access_token_jwks_urls` are specified in the APIGWD configuration file `/etc/rtbrick/apigwd/config.json`, it downloads the file from one of the specified urls.

The scope claim contains the roles the user has. For example:

```
{
  "sub": "1234567890",
```

```
  "name": "John Doe",
  "admin": true,
  "iat": 1516239022,
  "exp": 1600000000,
  "scope": "user operator"
}
```

This user has the roles user and operator.

#### JWKS Validation

The example below shows how to create a public/private key set file (for example with https://mkjwk.org/)

*Figure 153. JWKS Validation*

Example PubPrivJwks.json:

```
{
  "keys": [
    {
      "p": "-
3WNqU2aWKy8Q7mblRpw_aOknW47YvZKVNQzlWdijXW5E1hQ5c6sfjqo92pq5mNKJ_Xkl71xHyp-WBfw-
xJqZ9pUuG4jnUiKgzYHvkccDF5XIMrpA67VnBozmyLckQOKEXesRD2hacrjb-
T89dcIZQHBUK1RYXGRxHCM_hPeBok",
      "kty": "RSA",
      "q": "07STzbuUh6p_iN2wzTegIQdnXBLnzPObCKt-KLSjkLtrZMv2YxM2yhMs-
56SsLR7EICFRAB2vdCzlovXKShubU9NSKgVI38qpGV9hii8rqN5N3dEM4Gscp_TR36ex1NoSC6kHBF2hsyPfgD
7U-txguZr6w9MN6rK4bAcg0TWGM0",
      "d": "IWcC4aKuPVlVGZeBhb3mla1mOsOcJuLxjpkZAu-
QaQO7phWzeGLEfln3tojKtIK6e11FEz137ow75Je0_oMAzE-
eTAMyseUTHZhn4LhmIdOwnsp9OZrMbNmLFpaF_rGYb0630xg27GdR4gC_lZvMuy1uCP1sr8Iyl1ujN7n_6ETMY
bdXPLOtEAB4Dag6XFTjy1l8aVvBxpscm8gKg64fgBGRvGY6PiUfqY3gaq_vX9SOOHVPN5WoShKA4fguxukRBiL
LYNxDMDb3-h8pd1_fr2WayDzmcIXpxvVjRVZHt71C-0Uhap6eRDMQocZXih8IdNV8zHUF_LeciE36fIb3oQ",
      "e": "AQAB",
      "use": "sig",
      "kid": "access",
      "qi": "3His_DaBkf_r7uDx9-8BOhOQPhcudT95XC9WyS5MrYIBtgqQi6IscHIqvtXFpjmPRey-
chO7p9msOAB_T8j_mg1l6UWOx6j4h_fyHEbOwRqfNemKng2Hs0uCrwpjgGf2eXzaBY8T9HlbFlTJAAARGh_PeP
Bi-F-IfAxGayj4hiM",
      "dp": "NJuYYpZAt1KUJJsdSKl6gCYPV3xrYj3iuTKYBCbYAH5jlP-
CFUIS5mnBVdnmuYKGTivsgi55DysluapwmSZ2KnoMBXXNb6dwixjvr8hSvuex1MN-
0m1udTUqHMfDW3dhGFxwJuq57VcsFAnVPl2ZfQBMAGPyRa-r7mwZo0Jmzfk",
      "alg": "RS256",
      "dq": "XL-
```

```
4IWIU6Hrh9OxrEP1VwiKkPcpqk3gGa_31_49kOXxiyH4zK6S3VECibHpEefYYFFq6B9jMLMzKYSJS2U1FU85yZ
Wp-GFcWL3_nRmeCgmBMMuilkIs3KeCrh58JoPoBrd4BN-rOqq_kDagQc-uqh1a74PeKxLimucmWNExsH-E",
      "n": "z_NDmLu8M3KGvxvfJt8CAhdLdsqkskfY7vf9X9pW1LE_r31_HU85-
l6NNHeUWYbSNe6lt9YODnL8-
vTT6oCgre96byvpdYZ7Ki5KGe4fU96x0_ZF5LceUQc4l5dx6aptNi9mWgcZ9nkc2Xh83ASg9otG2YoYsAnI1cO
0TjzV9cMI7u7VON6SON9wbWFY01--ixMqxRAZuEJjbg4QAdL7DndRQXvmq1m7lv-
nnPPQ0a7ZTg7NZDEn5lMmadUlTVl5uvSNsACtC49R5kEkNCc1Hc-
3gootU5VyVPBx6IFHtNC2BiGasQAUpsDXZl7YtvBZwzYZwznUlluPiKLDk-4TtQ"
    }
  ]
}
```


The APIGWD only needs to know the Public part:


```
{
 "keys": [
 {
     "kty": "RSA",
     "e": "AQAB",
     "use": "sig",
     "kid": "access",
     "alg": "RS256",
     "n": "z_NDmLu8M3KGvxvfJt8CAhdLdsqkskfY7vf9X9pW1LE_r31_HU85-
l6NNHeUWYbSNe6lt9YODnL8-
vTT6oCgre96byvpdYZ7Ki5KGe4fU96x0_ZF5LceUQc4l5dx6aptNi9mWgcZ9nkc2Xh83ASg9otG2YoYsAnI1cO
0TjzV9cMI7u7VON6SON9wbWFY01--ixMqxRAZuEJjbg4QAdL7DndRQXvmq1m7lv-
nnPPQ0a7ZTg7NZDEn5lMmadUlTVl5uvSNsACtC49R5kEkNCc1Hc-
3gootU5VyVPBx6IFHtNC2BiGasQAUpsDXZl7YtvBZwzYZwznUlluPiKLDk-4TtQ"
   }
 ]
}
```


Now to create a token you can use https://keytool.online/, and paste the PubPrivJwks.json into the RSA Key Field and provide as Payload.

For example:

```
{
  "sub": "1234567890",
  "scope": "user operator",
  "name": "John Doe",
  "admin": true,
  "exp": 1600000000,
  "iat": 1516239022
}
```

This results in the following token:

```
eyJraWQiOiJhY2Nlc3MiLCJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwic
2NvcGUiOiJ1c2VyIG9wZXJhdG9yIiwibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWUsImV4cCI6MTYwMDA
wMDAwMCwiaWF0IjoxNTE2MjM5MDIyfQ.mP0mXR96-
9gYIzh6_2saUQckKDwpC7jDFpo2m0g9YAj4DkSf4xDoxBqMRwFkntLC6NV0sxyUNzC-
nv5yBretJAbbX_hCMS5Jk392piCVMt9ucbwnCKs6xaJDJmMHI1qxyf7lCgd9nIlawme4_nQnMJ4N9RdVeIuyv1
siuNUOo9RdSE4cX2JIzlrjgoZmtcU-
nq_I7S2QTkdro2e1wPZKktTMAoG6VjGb7ieIQ5XyLKNQt9PWSZ2sHkd85MxXMRUWUcrEagW6JrV3uixeT3QTZ3
g9Y6Qb4XDPH3EXUoAHJ0V26rpqXDsB_nmNvI5CVvCUcaZLPYoSEzBUPa9NaFIBcg
```


The `apigwd` can decode that token and validates the token with the corresponding key in the specified JKWS file.

### 9.1.3. Role Based Access Control (RBAC)

Role Based Access Control (RBAC) is an approach to restrict the system access to authorized users. The authorization model is role-based. There will be three items in a role-based modeel: `sub`, `obj`, and `act`.

- **sub**: the user (role) that wants to access a resource.
- **obj**: the resource that is going to be accessed
- **act**: the operation that the user performs on the resource

The RBAC Data Model is implemented in RBFS, and it allows you to define Permission or User Roles to various type of resources.

The model contains:

- **Resource Type**: The type of resource we are talking about (for example, BDS Table, BDS Object, REST)
- **Resource**: The identifier of the Resource (for example, Table Name, Rest endpoints). Regular expressions are allowed.
- **Permissions**: Indicates the action that a user is allowed to perform on the resource. The Permissions are CRUD (Create, Read Update, Delete). The permission gets a semantic with respect to the resource type.
- **Role**: The role of a user who tries to access a resource.

#### CTRLD Authorization Configuration

##### Activate or Deactivate Authorization in CTRLD

```
"auth_disabled": true
```

It is possible to specify the permissions in CTRLD exactly in the way specified above.

Where sub is the role a user needs to have, obj species the url endpoint the user wants to reach, and act is the HTTP Method the user wants to call on the endpoint.

For example:

```
{
 "permissions": [
   {"sub": "supervisor", "obj": "/*", "act": ".*" },
   {"sub": "reader", "obj": "/*", "act": "GET"},
   {"sub": ".*", "obj":
"/api/v1/rbfs/elements/\{element_name}/services/\{service_name}/proxy/*", "act": ".*"}
 ]
}
```

- The user with the role supervisor is allowed to access all rest endpoints, and act on them with all HTTP methods.
- The user with the role reader is allowed to access all rest endpoints, but can only call the HTTP GET method.
- All authenticated users are allowed to access the proxy endpoint with all HTTP methods.

To configure that policy CTRLD offers 2 endpoints:

- PUT `/api/v1/ctrld/authorization/permissions`
- GET `/api/v1/ctrld/authorization/permissions`

Please refer to API Documentation for more information.

### RBFS Authorization configuration

#### RBFS Role Configuration via REST

```
{
 "objects": [
   { "attribute": { "role": "operator", "permission": "create|read|delete",
"resource_regex": "global.*", "resource_type": "object" } },
   { "attribute": { "role": "operator", "permission": "create|read|delete",
"resource_regex": "global.*", "resource_type": "table" } }
 ],
 "table": { "table_name": "secure.global.rbac.authorization.config", "table_type":
"authorization_config_table" }
}

{
 "objects": [
   { "attribute": { "role": "user", "permission": "-|read|-", "resource_regex":
"global.*", "resource_type": "table" } },
   { "attribute": { "role": "user", "permission": "-|read|-", "resource_regex":
"global.*", "resource_type": "object" } }
 ],
```

```
 "table": { "table_name": "secure.global.rbac.authorization.config", "table_type":
"authorization_config_table" }
}
```

- **role** : Represents role in the system
- **resource_type** : Represents resources in the RBFS (table|object).
- **resource_regex** : Regex for the resources to be accessed.
- **permission** : Bitmap representing permissions to create, read and delete. create|read|delete

| Action | BDS Table | BDS Object |
|---|---|---|
| Create | Create a BDS Table | Create/Update a BDS Object |
| Read | Read Table Header Objects or Metadata | Read BDS Objects |
| Delete | Delete a BDS Object | Delete a BDS Object |

#### RBFS Authorization CLI Configurations

**Global user role configuration**:

**set system authorization global role** <name> **rbac-permission** <resource-type> <resource-regex> **permission** <permission-map>

| | |
|---|---|
| `role` | Represents role in the system |
| `resource_type` | Represents resources in the RBFS (table/object). |
| `resource_regex` | Regex for the resources to be accessed. |
| `permission` | Bitmap representing permissions to create, read and delete.<br><br>-/-/-<br><br>-/-/delete<br><br>-/read/-<br><br>-/read/delete<br><br>create/-/-<br><br>create/-/delete<br><br>create/read/-<br><br>create/read/delete |

**Example**

```
admin@rtbick: cfg> set system authorization global role admin rbac-permission table
global.* permission create/read/delete
```

**Lawful user role configuration**

**set system authorization lawful role** <name> **rbac-permission** <resource-type> <resource-regex> **permission** <permission>

| | |
|---|---|
| `role` | Represents lawful interceptor (LI) role in the system |
| `resource_type` | Represents resources in the RBFS (table/object). |
| `resource_regex` | Regex for the resources to be accessed. |
| `permission` | Bitmap representing permissions to create, read and delete.<br><br>-/-/-<br><br>-/-/delete<br><br>-/read/-<br><br>-/read/delete<br><br>create/-/-<br><br>create/-/delete<br><br>create/read/-<br><br>create/read/delete |

**Example**

```
admin@rtbick: cfg> set system authorization lawful role fbi rbac-permission table
local.* permission -/read/-
```

### 9.1.4. SSH with TACACS+

RBFS provides a custom pluggable authentication module that gets invoked by the stock sshd on login. The necessary configurations are pre-installed on RBFS.

RtBrick-PAM, referred to as RTB-PAM helps in landing the TACACS authentication on the appropriate user in the Ubuntu container and helps in providing necessary details for the secure management plane feature.

Once the PAM client requests TACACS for the authentication, with successful authentication TACACS

responds with a few RtBrick specific details.

```
{
     rtb-deny-cmds: clear bgp peer
     priv_lvl : some_level
}
```

On successful authentication, the RTB-PAM module creates a token (JWT) for the logged-in ssh user.

#### RTB-PAM Token

Token created by the RTB-PAM module contains the same claims that are defined under the RtBrick Token section, and this token is signed with the `secret_jwks.json` key.

The `scope` claim in the `rtb-token` is derived from the Linux groups that the locally mapped user belongs to.

The deny commands are converted into the claim `rtb-deny-cmds`. Once the token is created, it is transferred to the environment variable.

```
setenv RTB_TOKEN = {
  "sub": "83692",
  "iat": 1516239022,
  "exp": 1517239022,
  "name": "Admin User",
  "preferred_username", "user1",
  "scope": "operator tacacs_priv_lvl_8"
  "rtb-deny-cmds": "^clear bgp peer"
}
```

After the RTB-PAM token is created, the CLI prompt appears. If a token is not created for the logged-in user, then the user cannot perform communication with the BD.

#### SSH User Prompt

After you successfully log into RBFS via SSH, you can see the `rtb-token` using the shell environment. For example, an SSH prompt may look like the example below.

```
rtbng@b908f71f63b7:~$ env
SSH_CONNECTION=198.51.100.1 33136 198.51.100.44 22
LESSCLOSE=/usr/bin/lesspipe %s %s
LANG=C.UTF-8
USER=rtbng
PWD=/home/tacacs12
HOME=/home/tacacs12
SSH_CLIENT=198.51.100.1 33136 22
SUDO_USER=rtbng
PRIV_LVL=1
```

```
SSH_TTY=/dev/pts/1
SUDO_PROMPT=[sudo] password for rtbng:
MAIL=/var/mail/rtbng
TERM=xterm-256color
SHELL=/bin/bash
SHLVL=1
LOGNAME=rtbng
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/loca
l/games
LESSOPEN=| /usr/bin/lesspipe %s
_=/usr/bin/env
RTB_TOKEN=eyJhbGciOiJIUzI1NiIsImtpZCI6InJ0YnJpY2siLCJ0eXAiOiJKV1QifQ.eyJleHAiOjE0NTE2M
DczMDAsImlhdCI6MTQ1MTYwNjQwMCwiaXNzIjoicnRicmljay1hcGktZ3ciLCJuYW1lIjoiQmhpc2htYSBBY2h
hcnlhIiwicHJlZmVycmVkX3VzZXJuYW1lIjoiYmhpc2htYSIsInNjb3BlIjoic3lzdGVtIiwic3ViIjoiOTllO
GI0YTEtM2E2Yi00YzI5LWJlZGItN2U3N2NjOTFjZTZiIn0.NOOAcafmHfgx-QFwiC-
_VGokbvUwrTOjhfpD9px3hMY
```

The users having access to the Linux shell only can see the installed rtb-token in the shell environment.

**User Login Flow**

The figure below shows the user login flow.

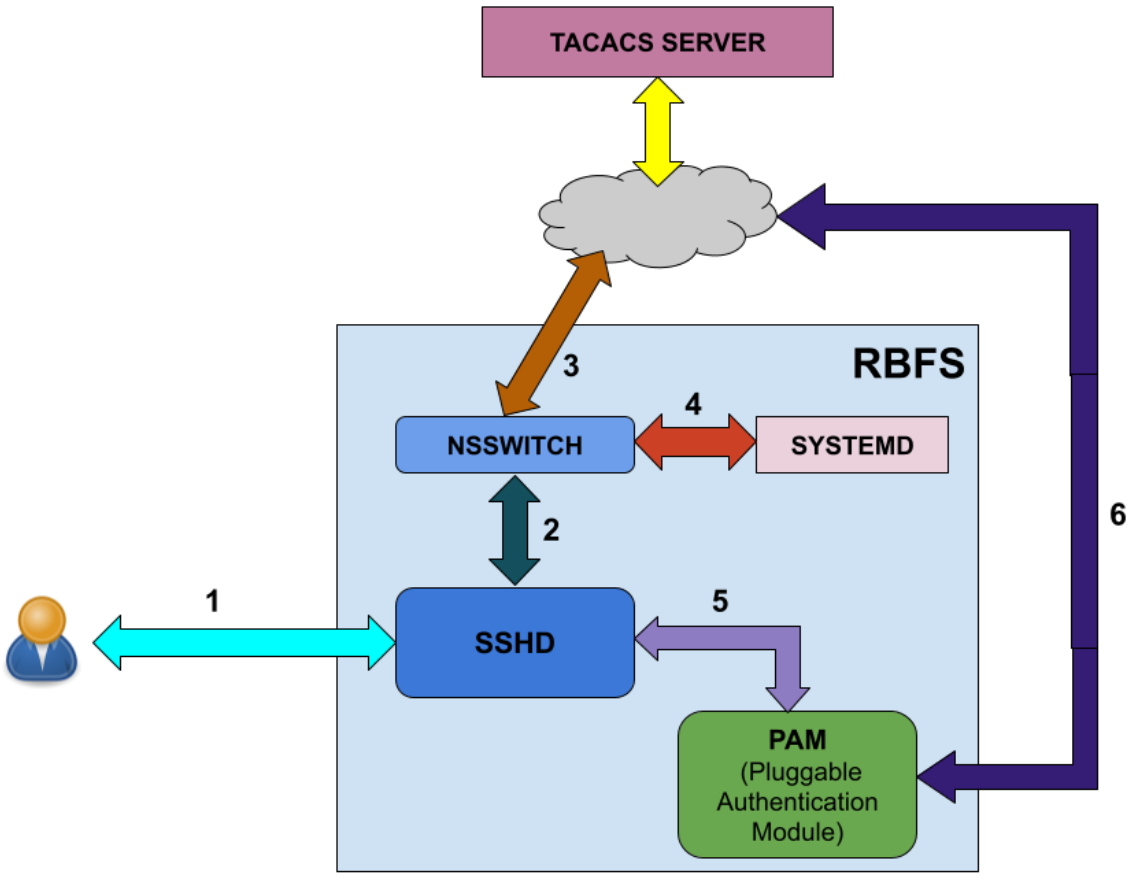



*Figure 154. User Login Flow*

As shown in the figure above, the following steps are involved in the user login flow.

1. User starts to initialize using ssh
2. The SSH daemon (sshd) refers to the Name Service Switch (NSSWITCH) for user lookup
3. NSSWITCH performs a user lookup at the TACACS server

> A user lookup happens at TACACS user database even for the local users, except for the user 'supervisor'.

4. If the user is not found at TACACS user database, then the user look up happens at the local user database.
5. User authentication is performed by PAM modules
6. PAM communicates to the TACACS server for authentication if it is a TACACS user

**Linux pre-configured users and groups**

| User Name | Group Name | Privilege |
|---|---|---|
| supervisor | supervisor | level 15 |
| operator | operator | level 7-14 |

| User Name | Group Name | Privilege |
|---|---|---|
| reader | reader | level 0-6 |

### In-Band and Out-of-Band TACACS user SSH Login

#### In-band TACACS user SSH Login

A TACACS user can login using SSH to rtbrick container through inband management. RBFS should be configured with inband-management TACACS server for TACACS user login, and inband management configuration should be enabled with TACACS service.

#### Out-of-band TACACS user SSH Login

TACACS user can login via SSH to RtBrick Host though out-of-band management. RBFS should be configured with out-of-band management TACACS server for TACACS user login.

### Configuring TACACS+ for RBFS

To configure TACACS+ server for RBFS, enter the the following commands.

### Syntax

**set system authorization tacacs** <ipv4-address> <attribute> <value>

### Command Arguments

| Attribute | Description |
|---|---|
| <ipv4-address> | IPv4 address of the TACACS+ Server. |
| <inband \| out-of-band> | Specifies inband or out-of-band connectivity to the TACACS+ server. |
| <inband \| out-of-band> port | Specifies the TACACS+ server's port. |
| <inband \| out-of-band> secret-encrypted-text <secret-encrypted-text> | Specifies the TACACS+ shared secret (encryption) key with a length of 4 to 16419 characters. |
| <inband \| out-of-band> secret-plain-text <secret-plain-text> | Specifies the TACACS+ secret (encryption) key in plain text format with a . Specifies the TACACS+ global passkey in plaintext format with a length of 1 to 240 characters. |

For TACACS login to work with inband management, inband configuration must be enabled with the following command:
**set inband-management instance** <instance-name> **tacacs true**

### Example

```
root@rtbrick: cfg> set system secure-management-status true
```

```
root@rtbrick: cfg> set system authorization tacacs 192.0.2.130 inband secret-plain-
text RtBrick_Little_Secret
root@rtbrick: cfg> set inband-management instance inband_mgmt tacacs true
```

A TACACS user is not allowed to login without TACACS+ server configuration.

The example below shows the running configuration after you configure TACACS+.

```
{
  "rtbrick-config:system": {
    "authorization": {
      "tacacs": [
        {
          "ipv4-address": "192.0.2.130",
          "type": "inband",
          "secret-plain-text": "RtBrick_Little_Secret"
        }
      ]
    }
  }
}
```

**Example: TACACS User Configuration in the TACACS Server**

The example below shows the server configurations for `rtb-deny-cmds`.

To collect debug information, the `linux-group="debug_info,rtbrick_confd_sock_access` option should be specified.

```
accounting file = /var/log/tac_plus.acct
key = tacacskey

user = bob {
    login = cleartext "bob"
    member = Network_Operator
}
group = Network_Operator {
  default service = permit
  service = exec {
        priv-lvl = 10
        rtb-deny-cmds = "show bgp.*"
      linux-groups = "debug_info,rtbrick_confd_sock_access"
  }
}
```

The `priv-lvl` is a mandatory attribute in the TACACS user configuration. The `priv-lvl` from 0 to 14 always initiate with "/usr/local/bin/cli" and `priv-lvl` 15 starts with

> the Linux shell "/bin/bash".

Multiple cmd-regexes can be configured with each regexes separated by semicolon (;).

Example:

```
rtb-deny-cmds = "show bgp .*;show isis .*"
```

#### Troubleshooting NSS User Lookup Issues

To look up the TACACS username with all NSS methods, enter the following command:

```
ubuntu@rtbrick:~$ sudo getent passwd <tac_user>
```

To look up the local user within the local user database, enter the following command:

```
ubuntu@rtbrick:~$ sudo getent -s compat passwd <local_user>
```

To look up the TACACS user within the TACACS+ server database, enter the following command:

```
ubuntu@batman:~/development$ sudo getent -s tacplus passwd <tacuser>
```

If TACACS does not appear to be working correctly, You can enable debug logging by adding the `debug=1` parameter to one or more of these files:

```
/etc/tacplus_servers
/etc/tacplus_nss.con
```

### Configuring Secure Management Logs

#### SSH User Login Logs

The transaction logs of users (in the PAM module) are available in the following log file:

```
/var/log/auth.log
```

Commands to enable secure-management logging:

**set log bd** <bd_name> **module secure_management logmap** <log_map> **level** <log-level>

**Example**:

```
root@rtbrick: cfg> set log bd all module secure_management logmap all level Info
```

# 9.2. Securing Control Plane

## 9.2.1. Securing the Control Plane Overview

Control plane security is essential for ensuring the reliable operation of a router by controlling the flow of traffic directed from the forwarding plane to the control plane. This protection is achieved primarily through filtering or rate-limiting mechanisms that prevent unwanted or potentially harmful traffic from overwhelming the router's control functions. In RBFS, the control plane is secured by using Access Control Lists (ACLs) and policers, which are implemented and enforced on the forwarding plane. These ACLs and policers allow precise control over which packets are allowed to reach the control plane and at what rate, effectively safeguarding essential router processes from excessive load or attacks.

Furthermore, control plane security includes the capability to mark traffic generated by the control plane itself. This traffic marking facilitates prioritization within the network, enabling Quality of Service (QoS) benefits and allowing other network devices to identify and apply appropriate filtering or handling policies.

By combining filtering, rate-limiting, and traffic marking, RBFS provides a comprehensive framework to maintain the integrity, stability, and performance of the router's control plane in complex network environments.

**Control Plane Traffic**

The control plane handles all routing, access, and management protocols. It processes information and configures it into the forwarding plane. For example, routes learned via protocols such as BGP and chosen by the route resolver are programmed into the forwarding plane.

An exception is the management protocols running on the out-of-band management port (typically named me0), which is directly linked to the control plane. Traffic received on this interface does not traverse the forwarding plane and thus is not protected by control plane security mechanisms. This separation ensures no forwarding between forwarding plane interfaces and the out-of-band management interface in the control plane.

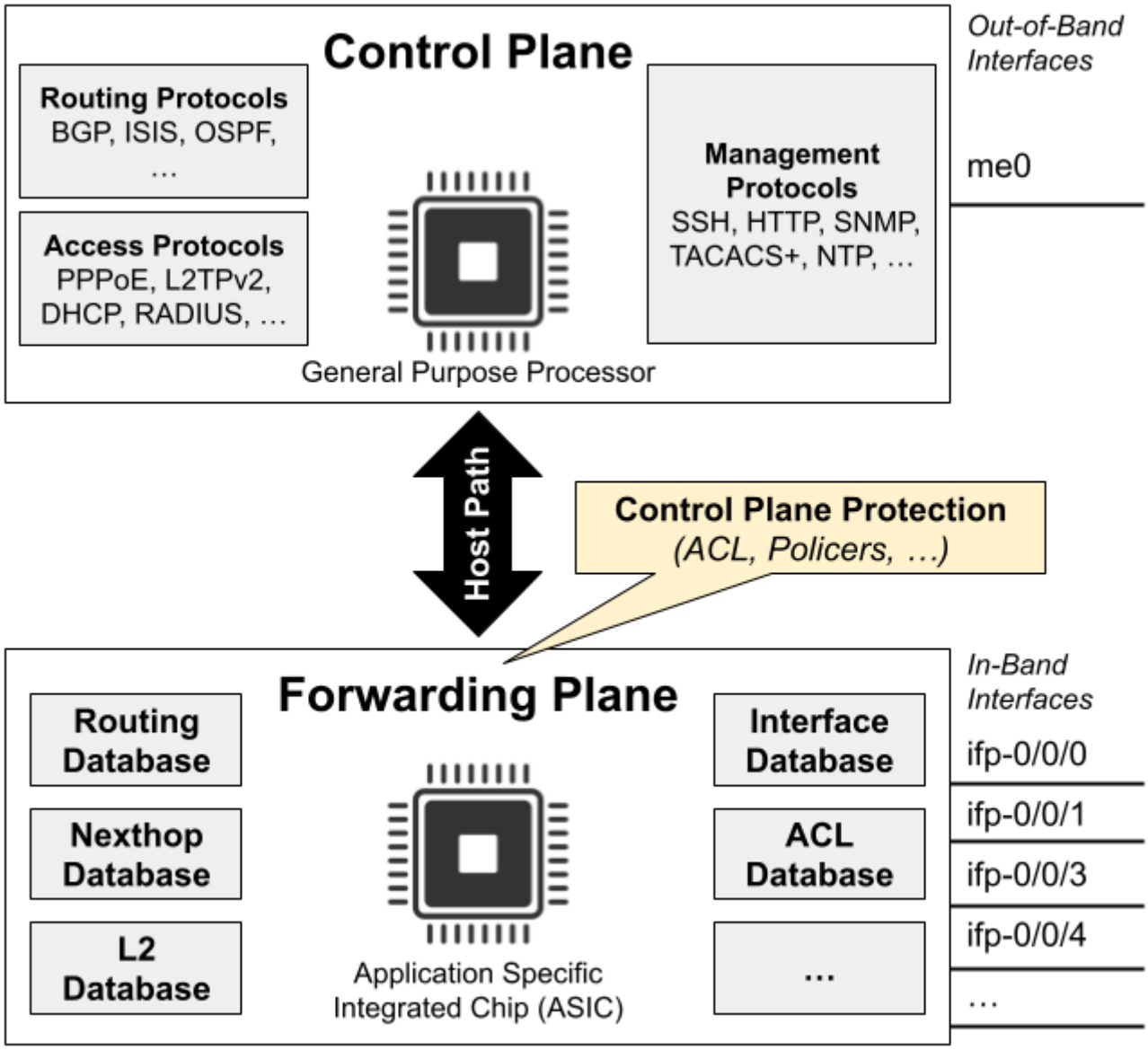



*Figure 155. Control Plane Traffic*

The diagram shows routing protocols (BGP, ISIS, OSPF), management protocols (SSH, HTTP, SNMP, TACACS+, NTP), and access protocols (PPPoE, L2TPv2, DHCP, RADIUS) associated with the control plane. The control plane uses general-purpose processors and builds databases like forwarding and ACL tables.

Conversely, the forwarding plane uses copies of these tables to make packet flow decisions based on match-action entries. The forwarding plane is implemented with high-performance ASICs capable of handling very high packet rates.

There are two kinds of traffic:

1. **Control Traffic**: Packets destined to the device itself, processed by the router CPU. These are identified by matching destination address, ACL rules, or exceptions (e.g. invalid fields, packet size, TTL).
2. **Transit Traffic**: Packets passing through the router, not destined to it, are forwarded via physical interfaces.

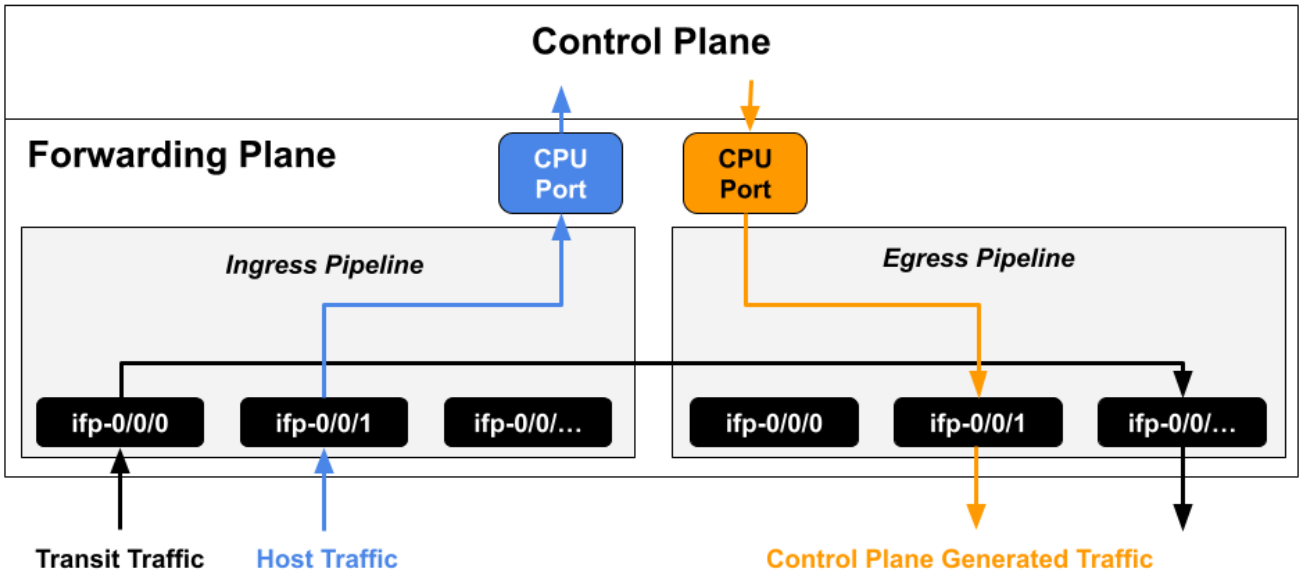



*Figure 156. Traffic Types*

All control traffic is directed to CPU ports for processing by the control plane. Because general-purpose processors are not optimized for high packet rates, excessive control plane traffic (e.g. from an attacker) can overwhelm the router. Hence, controlling, filtering, and rate-limiting unwanted control-plane traffic is essential for protection.

### Securing the Control Plane

In RBFS, control-plane traffic can be redirected to the CPU by three main mechanisms:

1. via ACLs
2. via route lookup
3. via exception (e.g. packet size, TTL)

All those mechanisms need to be considered and secured separately, as described in the following sections.

#### Control Plane Traffic via Protocol ACLs

Control plane security is built upon automatically generated Access Control Lists (ACLs) for enabled routing, management, and access protocols. These protocol ACLs direct their respective traffic to the control plane. Each ACL rule specifies match criteria and corresponding actions, such as redirecting traffic to the CPU and applying policers, when a packet matches.

> Such protocol ACLs are also sometimes referred to as `Trap`.

Protocol ACLs:

1. auto-generated
2. specific (e.g. match IPv6 link-local neighbors)
3. high priority of 50 (lower values have priority over higher values)
4. can be verified with `show acl` command

5. do not require manual configuration

Example 1: Protocol ACL created by BGP

```
supervisor@rtbrick: op> show acl detail
<...>
Rule: default_bgp_l4_trap_192.0.2.101_192.0.2.102_dst
  ACL type: l3v4
  Ordinal: -             Priority: 50
    Match:
      Direction: ingress
      Destination IPv4 address: 192.0.2.102
      Source IPv4 address: 192.0.2.101
      Destination L4 port: 179
      IP protocol: tcp
    Action:
      Redirect to CPU: True
      Policer profile name: _DEFAULT_POLICER_50_MB
    Result:
      Trap ID: bgp
      ACL Handle: 268
<...>
```

The default policer, `_DEFAULT_POLICER_50_MB`, restricts traffic on these ACLs to 50 Mbps. This bandwidth is shared across all protocols that reference this policer. While almost all protocols share the same policer by default, it is possible to configure dedicated policers for individual protocols or groups of protocols as described in the Control Plane Security Configuration section.

#### Support for Per-Protocol Control Plane Security Policers

RBFS provides control plane security policers for each protocol. It assigns a dedicated policer to each protocol. Protocols such as LLDP, IS-IS, and PPPoE rely on a independent control plane security policer. Each protocol receives its own default policer, ensuring fair resource allocation.

The following list provides newly added policers per protocol.

```
_DEFAULT_POLICER_LLDP_GLOBAL_10MB
_DEFAULT_POLICER_ARP_GLOBAL_10MB
_DEFAULT_POLICER_OAM_GLOBAL_20MB
_DEFAULT_POLICER_LDP_GLOBAL_50MB
_DEFAULT_POLICER_PPPOE_GLOBAL_50MB
_DEFAULT_POLICER_OSPF_GLOBAL_50MB
_DEFAULT_POLICER_ISIS_GLOBAL_50MB
_DEFAULT_POLICER_BGP_GLOBAL_100MB
```

#### Control Plane Traffic via Route Lookup

By default, traffic sent to a router's IP address (referred to as "my IP") that lacks an existing Access Control List (ACL) will be forwarded to the CPU via route lookup. This behavior applies to IP

addresses configured on any interface, including loopback interfaces. To protect the control plane from malicious traffic targeting these IP addresses, user-defined ACL rules must be configured in RFBS. These are ACLs that are manually created by the user.

Configuring such ACLs to protect "my IP" is described in the Control Plane Security Configuration section.

#### Policer Support for TTL=1 Trap Packets

RBFS supports rate limiting for TTL=1 traffic handled by the control plane to prevent control plane overloading. By default, this feature is enabled with a default rate of 10Mbps.

### Limitations and Notes

The control-plane security features are supported on hardware platforms only. They are not supported on virtual deployments.

### Supported Platforms

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

## 9.2.2. Control Plane Security Configuration

### Secure Control Plane Traffic via Protocol ACLs

This section details the configuration options for control plane traffic that is redirected to the CPU using protocol ACLs. These ACLs are generated automatically by the protocols and do not require, nor can they be, manually configured. For instance, configuring a routing protocol such as BGP automatically creates the necessary ACLs to match BGP packets and forward them to the control plane.

#### Configuring the Control Plane Security Feature

Control plane security is active by default, utilizing the `_DEFAULT_POLICER_50_MB` to cap traffic across most of the protocol ACLs at 50 Mbps. This bandwidth is shared among all protocols using this policer. While protocols share a single policer by default, separate policers can be configured for specific protocols or groups.

Although enabled by default, control plane security can be deactivated, allowing all control plane traffic to be sent to the CPU without any rate limits.

```
set forwarding-options control-plane-security state disable
```

Disabling control-plane security is not recommended.

#### Configuring Protocol ACL Options

This section describes how to configure policers per protocol and configure a match on IPv4 ToS or

IPv6 TC fields for protocol ACLs.

**Syntax:**

**set forwarding-options control-plane-security protocol** <protocol-name> <attribute> <value>

| Option | Description |
|---|---|
| `protocol <protocol-name>` | Name of the protocol. You can configure individual protocols or all protocols using the `all` keyword. The protocol-specific configurations will take priority over `all` configuration.<br><br>Supported protocol options: `bgp`, `ospf`, `isis`, `radius`, `pppoe`, `ipoe`, `pim`, `igmp`, `user-defined`, `lldp`, `l2tp`, `inband`, `ldp-discovery`, `ldp-nbr`, `lag`, `rd`, `oam`, `all`. |
| `match-tc <tc-value>` | Configure IPv6 TC value. The range is 0 to 254. |
| `match-tos <tos-value>` | Configure IPv4 ToS value. The range is 0 to 254. |
| `policer <policer>` | Configure the policer name. |

While all protocols share the same policer by default, it is possible to configure dedicated policers for individual protocols or groups of protocols. Protocols referencing the same policer identified by name will share the same bandwidth limit. If you require separate policers for individual protocols or groups of protocols, even with identical rates, you must assign each a unique name.

Example policer configuration for BGP with 100 Mbps:

```
set forwarding-options class-of-service policer bgp-global
set forwarding-options class-of-service policer bgp-global flags color-blind
set forwarding-options class-of-service policer bgp-global level1-rates cir 100000
set forwarding-options class-of-service policer bgp-global level1-rates cbs 10000
set forwarding-options class-of-service policer bgp-global level1-rates pir 100000
set forwarding-options class-of-service policer bgp-global level1-rates pbs 10000
set forwarding-options class-of-service policer bgp-global levels 1
set forwarding-options class-of-service policer bgp-global type two-rate-three-color
set forwarding-options class-of-service policer bgp-global mode single-level
```

Assign the policer to BGP protocol ACL:

```
set forwarding-options control-plane-security protocol bgp policer bgp-global
```

Other protocols can be similarly protected by either assigning a dedicated policer to each protocol or grouping several protocols under a single policer. We recommend a maximum of 200 Mbps for any single policer, a value that might only be necessary for BGP on certain peering routers. For all other protocols, up to 50 Mbps is typically sufficient.

To enhance system protection, the `match-tos` and `match-tc` options can be used to filter protocol traffic, ensuring only correctly marked packets are processed. This adds a layer of security by assuming an attacker would be unable to send traffic with these specific markings.

```
# accept BGP only with TOS 192
set forwarding-options control-plane-security protocol bgp match-tc 192
# send BGP with TOS 192
set forwarding-options class-of-service control-plane-qos outbound-marking protocol bgp tos codepoint 192
```

### Secure Control Plane Traffic via Route Lookup

This section details the configuration for control plane traffic that is redirected to the CPU via route lookup, which refers to traffic sent to a router's IP address (referred to as "my IP") that lacks an existing Access Control List (ACL). This behavior applies to IP addresses configured on any interface, including loopback interfaces. By default, this type of traffic is neither restricted nor rate-limited.

#### Configuring ACLs

In RFBS, ACLs are applied globally and do not require explicit attachment. Rule names serve as configuration handles and do not need to be referenced. However, it is advisable to use distinct rule names for different topics, as counter names are derived from them. For example, use "myip-icmp" and "myip-icmpv6" for rules with dedicated counters.

To secure the control plane, you must apply ACLs and policers through configuration. You do not need to account for protocol traffic already handled by automatically generated protocol ACLs. Instead, you only need to define rules for other traffic sent to "my IP." In the simplest scenario, you can deny all other traffic. Typically, however, you will want to allow specific additional traffic, such as ICMP with a rate limit, while denying everything else. Alternatively, you could allow all traffic but apply a small policer rate.

**Syntax:**

**set forwarding-options acl** <options>

| Option | Description |
|---|---|
| `l3v4` | ACL configuration for IPv4 |
| `l3v6` | ACL configuration for IPv6 |
| `rule <rule-name>` | Name of the ACL rule |
| `ordinal <ordinal-value>` | Number of the configuration entry. Please note that the order of the configuration entries (ordinals) does not determine the processing. |

| Option | Description |
| --- | --- |
| `match <condition>` | Set match conditions such as source and destination prefixes, prefix lists, ports, and protocols. When multiple match conditions are specified within the same ordinal, they are combined as a logical AND, meaning all conditions must be met. |
| `action <action>` | Set action like permit, drop, or policer-name. |
| `priority <value>` | Configure the ACL entry priority. Lower values take precedence, with the default set to 500. RBFS utilizes TCAM-based ACLs to enable parallel evaluation of all rules, which allows for high-speed processing and efficient matching. When multiple rules match a given packet, the highest priority rule (lowest value) is applied. |

RBFS supports the match conditions `destination-ipv4-local` and `destination-ipv6-local`. These options are converted internally into a prefix list that includes all interface addresses. As a result, any rule using these match options is installed multiple times, once for each interface address.

The following rule is an example rule that drops all traffic with a destination one of the router IP addresses, but since the priority of 2000 is higher than the protocol priority of 50, the protocol traffic, like BGP, is not dropped, since the BGP protocol rule also matches and is prioritized.

```
set forwarding-options acl l3v4 rule myip-discard
set forwarding-options acl l3v4 rule myip-discard ordinal 2000
set forwarding-options acl l3v4 rule myip-discard ordinal 2000 match direction ingress
set forwarding-options acl l3v4 rule myip-discard ordinal 2000 match destination-ipv4-local true
set forwarding-options acl l3v4 rule myip-discard ordinal 2000 action drop true
set forwarding-options acl l3v4 rule myip-discard ordinal 2000 action statistics true
set forwarding-options acl l3v4 rule myip-discard ordinal 2000 priority 2000
```

When designing security ACLs to protect "my IP", you may don’t want to completely block all unwanted traffic. Instead, you might allow and rate-limit specific necessary traffic, such as ICMP. It’s important to remember that you do not need to consider protocol traffic already handled by the protocol ACL when creating these security ACLs.

In environments with thousands of logical interfaces and multiple rules, an alternative approach to reduce ACL usage could be employing custom prefix lists with aggregate prefixes of interface addresses or other match criteria. This is why these ACLs are not automatically generated, as each environment is unique and may require different rules. The configuration section provides common best practices that are applicable to many networks or can serve as a foundation for modifications based on specific requirements.

#### Restricting In-Band Management Access

Enabling in-band management, such as SSH, automatically creates protocol ACLs. These ACLs are configured to match the enabled protocols, directing management traffic to the control plane. By default, this traffic is not restricted by source IP addresses. To further enhance security, in-band management access can be restricted to trusted IP addresses by applying a source prefix list, which

adds an extra match condition to the in-band management protocol ACLs.

> This prefix list must contain /32 only! Any other prefix length is not supported!

```
set forwarding-options prefix-list trusted-ssh ipv4-prefix 10.10.10.1/32
set forwarding-options prefix-list trusted-ssh ipv4-prefix 10.10.10.2/32

set inband-management instance default ssh true
set inband-management instance default source-prefix-list trusted-ssh
```

By default, in-band management traffic is limited by control-plane-security to 20 Mbps using the `_DEFAULT_POLICER_20_MB` policer. This policer can be changed as previously explained using the protocol `inband`, as well in-band can also be optionally limited to only match if a given tos/tc marking is set.

```
set forwarding-options class-of-service policer inband-global flags color-blind
set forwarding-options class-of-service policer inband-global level1-rates
set forwarding-options class-of-service policer inband-global level1-rates cir 100000
set forwarding-options class-of-service policer inband-global level1-rates cbs 10000
set forwarding-options class-of-service policer inband-global level1-rates pir 100000
set forwarding-options class-of-service policer inband-global level1-rates pbs 10000
set forwarding-options class-of-service policer inband-global levels 1
set forwarding-options class-of-service policer inband-global type two-rate-three-color
set forwarding-options class-of-service policer inband-global mode single-level

set forwarding-options control-plane-security protocol inband policer inband-global
set forwarding-options control-plane-security protocol inband match-tos 128
```

#### Marking Outbound Control Plane Traffic

RBFS offers a flexible configuration of TOS, TC, or VLAN P-BIT markings for outbound control plane traffic.

These settings can be applied per protocol and, in some cases, individually per interface. While both Layer 2 (P-BIT) and Layer 3 (TOS) markings are configurable for all protocols, support for each marking type may vary, with some protocols supporting only one and others supporting both.

*Supported QoS types and IFP support*

| Protocol | Markings | IFP-based | IFL-based |
|---|---|---|---|
| l2-all | p-bit | NO | NO |
| l3-all | tos | NO | NO |
| BGP | tos | NO | YES |
| OSPF | tos | NO | YES |
| LDP | tos | NO | YES |

| Protocol | Markings | IFP-based | IFL-based |
|---|---|---|---|
| ISIS | p-bit | NO | YES |
| PIM | tos | NO | YES |
| DHCP Relay | p-bit | NO | YES |
| RADIUS | tos | NO | YES |
| IPoE | tos | NO | NO |
| PPPoE | tos/p-bit | YES(p-bit only) | NO |
| IGMP | tos/p-bit | YES(p-bit only) | NO |
| L2TP | tos | NO | YES |
| ARP/ND | p-bit | NO | YES |
| CFM | p-bit | NO | NO |
| MPLS-OAM | tos | NO | NO |

Settings are prioritized as follows: IFP/IFL (highest), protocol level, and l2-all/l3-all (lowest). For example, if a user configures the p-bit for both the ARP protocol and "l2-all," the ARP protocol setting takes precedence. If the ARP p-bit is then removed, the "l2-all" configuration will be applied.

Redundancy-related control traffic can be marked with a specific tos value with the L3-all option.

**Example:**

```
set forwarding-options class-of-service control-plane-qos outbound-marking protocol arp p-bit codepoint 6
set forwarding-options class-of-service control-plane-qos outbound-marking protocol arp p-bit ifl ifl-0/0/0/0 codepoint 5

set forwarding-options class-of-service control-plane-qos outbound-marking protocol l3-all tos codepoint 128

set forwarding-options class-of-service control-plane-qos outbound-marking protocol radius tos codepoint 192

set forwarding-options class-of-service control-plane-qos outbound-marking protocol ppp p-bit codepoint 6
set forwarding-options class-of-service control-plane-qos outbound-marking protocol ppp tos codepoint 192
```

The command `show qos control-plane` displays QoS remarking types, interface names, and codepoints for all or specified protocols.

```
op> show qos control-plane protocol ppp
Protocol        Remark Type       Interface         Codepoint
ppp             p-bit                  all              5
                                 ifp-0/0/1              3
                                 ifp-0/0/2              4
```

**Reference Configurations**

#### My-IP Discard

The following example demonstrates a basic configuration that should be suitable for many setups. It allows ICMP traffic but limits it to 10 Mbps using dedicated policers for ICMPv4 and ICMPv6. All other traffic not previously accepted by any protocol ACL is then discarded. If ICMP is not required, these rules can be removed.

```
set forwarding-options acl l3v4 rule myip-icmp ordinal 1500 match direction ingress
set forwarding-options acl l3v4 rule myip-icmp ordinal 1500 match ip-protocol icmp
set forwarding-options acl l3v4 rule myip-icmp ordinal 1500 match destination-ipv4-local true
set forwarding-options acl l3v4 rule myip-icmp ordinal 1500 action policer-name icmp-global
set forwarding-options acl l3v4 rule myip-icmp ordinal 1500 priority 1500

set forwarding-options acl l3v4 rule myip-discard ordinal 2000 match direction ingress
set forwarding-options acl l3v4 rule myip-discard ordinal 2000 match destination-ipv4-local true
set forwarding-options acl l3v4 rule myip-discard ordinal 2000 action drop true
set forwarding-options acl l3v4 rule myip-discard ordinal 2000 action statistics true
set forwarding-options acl l3v4 rule myip-discard ordinal 2000 priority 2000

set forwarding-options acl l3v6 rule myipv6-icmpv6 ordinal 1500 match direction ingress
set forwarding-options acl l3v6 rule myipv6-icmpv6 ordinal 1500 match ip-protocol ipv6_icmp
set forwarding-options acl l3v6 rule myipv6-icmpv6 ordinal 1500 match destination-ipv6-local true
set forwarding-options acl l3v6 rule myipv6-icmpv6 ordinal 1500 action policer-name icmpv6-global
set forwarding-options acl l3v6 rule myipv6-icmpv6 ordinal 1500 priority 1500

set forwarding-options acl l3v6 rule myipv6-discard ordinal 2000 match direction ingress
set forwarding-options acl l3v6 rule myipv6-discard ordinal 2000 match destination-ipv6-local true
set forwarding-options acl l3v6 rule myipv6-discard ordinal 2000 action drop true
set forwarding-options acl l3v6 rule myipv6-discard ordinal 2000 action statistics true
set forwarding-options acl l3v6 rule myipv6-discard ordinal 2000 priority 2000

set forwarding-options class-of-service policer icmp-global flags color-blind
set forwarding-options class-of-service policer icmp-global level1-rates
set forwarding-options class-of-service policer icmp-global level1-rates cir 10000
set forwarding-options class-of-service policer icmp-global level1-rates cbs 1000
set forwarding-options class-of-service policer icmp-global level1-rates pir 10000
set forwarding-options class-of-service policer icmp-global level1-rates pbs 1000
set forwarding-options class-of-service policer icmp-global levels 1
set forwarding-options class-of-service policer icmp-global type two-rate-three-color
set forwarding-options class-of-service policer icmp-global mode single-level

set forwarding-options class-of-service policer icmpv6-global flags color-blind
set forwarding-options class-of-service policer icmpv6-global level1-rates
set forwarding-options class-of-service policer icmpv6-global level1-rates cir 10000
set forwarding-options class-of-service policer icmpv6-global level1-rates cbs 1000
set forwarding-options class-of-service policer icmpv6-global level1-rates pir 10000
set forwarding-options class-of-service policer icmpv6-global level1-rates pbs 1000
set forwarding-options class-of-service policer icmpv6-global levels 1
set forwarding-options class-of-service policer icmpv6-global type two-rate-three-color
set forwarding-options class-of-service policer icmpv6-global mode single-level
```

#### Protocol Policers

This example demonstrates the configuration of user-defined policers. BGP traffic is limited to 100

Mbps, while ISIS and PPPoE traffic are each capped at 50 Mbps using dedicated policers.

```
set forwarding-options class-of-service policer bgp-global flags color-blind
set forwarding-options class-of-service policer bgp-global level1-rates
set forwarding-options class-of-service policer bgp-global level1-rates cir 100000
set forwarding-options class-of-service policer bgp-global level1-rates cbs 10000
set forwarding-options class-of-service policer bgp-global level1-rates pir 100000
set forwarding-options class-of-service policer bgp-global level1-rates pbs 10000
set forwarding-options class-of-service policer bgp-global levels 1
set forwarding-options class-of-service policer bgp-global type two-rate-three-color
set forwarding-options class-of-service policer bgp-global mode single-level

set forwarding-options control-plane-security protocol bgp policer bgp-global

set forwarding-options class-of-service policer isis-global flags color-blind
set forwarding-options class-of-service policer isis-global level1-rates
set forwarding-options class-of-service policer isis-global level1-rates cir 50000
set forwarding-options class-of-service policer isis-global level1-rates cbs 5000
set forwarding-options class-of-service policer isis-global level1-rates pir 50000
set forwarding-options class-of-service policer isis-global level1-rates pbs 5000
set forwarding-options class-of-service policer isis-global levels 1
set forwarding-options class-of-service policer isis-global type two-rate-three-color
set forwarding-options class-of-service policer isis-global mode single-level

set forwarding-options control-plane-security protocol isis policer isis-global

set forwarding-options class-of-service policer pppoe-global flags color-blind
set forwarding-options class-of-service policer pppoe-global level1-rates
set forwarding-options class-of-service policer pppoe-global level1-rates cir 50000
set forwarding-options class-of-service policer pppoe-global level1-rates cbs 5000
set forwarding-options class-of-service policer pppoe-global level1-rates pir 50000
set forwarding-options class-of-service policer pppoe-global level1-rates pbs 5000
set forwarding-options class-of-service policer pppoe-global levels 1
set forwarding-options class-of-service policer pppoe-global type two-rate-three-color
set forwarding-options class-of-service policer pppoe-global mode single-level

set forwarding-options control-plane-security protocol pppoe policer pppoe-global
```

### 9.2.3. Control Plane Security Operational Commands

#### Show Commands

This section describes operational commands available to verify various control-plane security features.

##### Verifying ACLs

The `show acl` command allows to verify protocol ACLs as well as user-defined ACLs.

**Syntax:**

**show acl** <options>

| Option | Description |
|---|---|
| `detail` | Displays all ACL details |
| `<acl-name>` | Displays the details for a single ACL |

Example 1: Protocol ACL with Control-Plane Security enabled

```
supervisor@rtbrick: op> show acl detail
Rule: _DEFAULT_MTU_POLICER_TRAP_ACL
  ACL type: l2
  Ordinal: -             Priority: 1
    Match:
      Attachment point: recycle-0/0/211
      Direction: ingress
      Physical Interface: recycle-0/0/211
    Action:
      Redirect to CPU: True
      Policer profile name: _DEFAULT_POLICER_20_MB
    Result:
      ACL Handle: 26
    Statistics:
      Units      Total       Accepted    Dropped
      Packets    -           -           -
      Bytes      -           -           -
Rule: recycle-0/0/215/1_bypass_acl
  ACL type: ext_l3v4
  Ordinal: -             Priority: 2
    Match:
      Direction: external
      Logical Interface: recycle-0/0/215/1
    Action:
      Stats enabled: True
    Result:
      Trap ID: user-defined
      ACL Handle: 3
      External:
        Value: 2367517
          Is Trap ID: Yes
          Trap ID: 14
          Trap Strength: 0
          Is QoS: Yes
          Class: 0
          Policer ID: 18
  ACL type: ext_l3v6
  Ordinal: -             Priority: 2
    Match:
      Direction: external
```

```
      Logical Interface: recycle-0/0/215/1
    Action:
      Stats enabled: True
    Result:
      Trap ID: user-defined
      ACL Handle: 6
      External:
        Value: 2498589
          Is Trap ID: Yes
          Trap ID: 14
          Trap Strength: 0
          Is QoS: Yes
          Class: 0
          Policer ID: 19
<...>
```

Example 2: User-defined ACL to Protect "my IP"

```
supervisor@rtbrick: op> show acl rule mpls.oam.destination.ipv4.trap.rule
Rule: mpls.oam.destination.ipv4.trap.rule
  ACL type: ext_l3v4
  Ordinal: -             Priority: 50
    Match:
      Direction: external
      Destination IPv4 prefix: 127.0.0.0/8
      Destination L4 port: 3503
      IP protocol: udp
    Action:
      Policer profile name: _DEFAULT_POLICER_OAM_GLOBAL_20MB
    Result:
      Trap ID: oam
      ACL Handle: 1
      External:
        Value: 77082513
          Is Trap ID: Yes
          Trap ID: 200
          Trap Strength: 7
          Is QoS: Yes
          Class: 0
          Policer ID: 588
```

#### Verifying ACL Counters

The `show acl statistics` command displays information about the ACL packet counters. The counters are useful to verify if the ACL rules actually match, and if potentially malicious traffic gets dropped.

**Syntax:**

## show acl statistics

Example: ACL statistics information

```
supervisor@rtbrick: op> show acl statistics
ACL                                       Ordinal    Type              Units      Total       Accepted    Dropped
lldp.ifp-0/1/32.trap.rule.0               -          l2                Packets    269         269         0
                                                                       Bytes      37153       37153       0
lldp.ifp-0/1/32.trap.rule.1               -          l2                Packets    269         269         0
                                                                       Bytes      37153       37153       0
lldp.ifp-0/1/32.trap.rule.2               -          l2                Packets    269         269         0
                                                                       Bytes      37153       37153       0
lldp.ifp-0/1/41.trap.rule.0               -          l2                Packets    269         269         0
                                                                       Bytes      37153       37153       0
lldp.ifp-0/1/41.trap.rule.1               -          l2                Packets    269         269         0
                                                                       Bytes      37153       37153       0
lldp.ifp-0/1/41.trap.rule.2               -          l2                Packets    269         269         0
                                                                       Bytes      37153       37153       0
lldp.ifp-0/1/42.trap.rule.0               -          l2                Packets    269         269         0
                                                                       Bytes      37153       37153       0
lldp.ifp-0/1/42.trap.rule.1               -          l2                Packets    269         269         0
                                                                       Bytes      37153       37153       0
lldp.ifp-0/1/42.trap.rule.2               -          l2                Packets    269         269         0
                                                                       Bytes      37153       37153       0
mpls.oam.destination.ipv6.trap.rule       -          ext_l3v6          Packets    1           1           0
                                                                       Bytes      56          56          0
mpls.oam.destination.ipv4.trap.rule       -          ext_l3v4          Packets    1           1           0
                                                                       Bytes      56          56          0
mpls.oam.source.ipv4.trap.rule            -          ext_l3v4          Packets    1           1           0
                                                                       Bytes      56          56          0
_DEFAULT_MTU_POLICER_TRAP_ACL             -          l2                Packets    -           -           -
                                                                       Bytes      -           -           -
lldp.ifp-0/1/46.trap.rule.0               -          l2                Packets    269         269         0
                                                                       Bytes      37153       37153       0
lldp.ifp-0/1/46.trap.rule.1               -          l2                Packets    269         269         0
                                                                       Bytes      37153       37153       0
lldp.ifp-0/1/46.trap.rule.2               -          l2                Packets    269         269         0
                                                                       Bytes      37153       37153       0
recycle-0/0/215/1_bypass_acl              -          ext_l3v4          Packets    174         174         0
                                                                       Bytes      12352       12352       0
recycle-0/0/215/1_bypass_acl              -          ext_l3v6          Packets    96          84          12
                                                                       Bytes      7957        7111        846
lldp.ifp-0/1/49.trap.rule.0               -          l2                Packets    269         269         0
                                                                       Bytes      37153       37153       0
lldp.ifp-0/1/49.trap.rule.1               -          l2                Packets    269         269         0
                                                                       Bytes      37153       37153       0
lldp.ifp-0/1/49.trap.rule.2               -          l2                Packets    269         269         0
                                                                       Bytes      37153       37153       0
lldp.ifp-0/1/97.trap.rule.0               -          l2                Packets    269         269         0
                                                                       Bytes      37153       37153       0
lldp.ifp-0/1/97.trap.rule.1               -          l2                Packets    269         269         0
                                                                       Bytes      37153       37153       0
lldp.ifp-0/1/97.trap.rule.2               -          l2                Packets    269         269         0
                                                                       Bytes      37153       37153       0
lldp.ifp-0/1/3.trap.rule.0                -          l2                Packets    269         269         0
                                                                       Bytes      37153       37153       0
lldp.ifp-0/1/3.trap.rule.1                -          l2                Packets    269         269         0
                                                                       Bytes      37153       37153       0
lldp.ifp-0/1/3.trap.rule.2                -          l2                Packets    269         269         0
                                                                       Bytes      37153       37153       0
```

```
lldp.ifp-0/1/47.trap.rule.0                -          l2                Packets    269          269          0
                                                                        Bytes      37153        37153        0
lldp.ifp-0/1/47.trap.rule.1                -          l2                Packets    269          269          0
                                                                        Bytes      37153        37153        0
lldp.ifp-0/1/47.trap.rule.2                -          l2                Packets    269          269          0
                                                                        Bytes      37153        37153        0
recycle-0/0/215_bypass_acl                 -          l2                Packets    -            -            -
                                                                        Bytes      -            -            -
lldp.ifp-0/1/16.trap.rule.0                -          l2                Packets    269          269          0
                                                                        Bytes      37153        37153        0
lldp.ifp-0/1/16.trap.rule.1                -          l2                Packets    269          269          0
                                                                        Bytes      37153        37153        0
lldp.ifp-0/1/16.trap.rule.2                -          l2                Packets    269          269          0
                                                                        Bytes      37153        37153        0
lldp.ifp-0/1/0.trap.rule.0                 -          l2                Packets    269          269          0
                                                                        Bytes      37153        37153        0
lldp.ifp-0/1/0.trap.rule.1                 -          l2                Packets    269          269          0
                                                                        Bytes      37153        37153        0
lldp.ifp-0/1/0.trap.rule.2                 -          l2                Packets    269          269          0
                                                                        Bytes      37153        37153        0
lldp.ifp-0/1/33.trap.rule.0                -          l2                Packets    269          269          0
                                                                        Bytes      37153        37153        0
lldp.ifp-0/1/33.trap.rule.1                -          l2                Packets    269          269          0
                                                                        Bytes      37153        37153        0
lldp.ifp-0/1/33.trap.rule.2                -          l2                Packets    269          269          0
                                                                        Bytes      37153        37153        0
lldp.ifp-0/1/96.trap.rule.0                -          l2                Packets    269          269          0
                                                                        Bytes      37153        37153        0
lldp.ifp-0/1/96.trap.rule.1                -          l2                Packets    269          269          0
                                                                        Bytes      37153        37153        0
lldp.ifp-0/1/96.trap.rule.2                -          l2                Packets    269          269          0
                                                                        Bytes      37153        37153        0
mpls.oam.source.ipv6.trap.rule             -          ext_l3v6          Packets    1            1            0
                                                                        Bytes      56           56           0
lldp.ifp-0/1/48.trap.rule.0                -          l2                Packets    269          269          0
                                                                        Bytes      37153        37153        0
lldp.ifp-0/1/48.trap.rule.1                -          l2                Packets    269          269          0
                                                                        Bytes      37153        37153        0
lldp.ifp-0/1/48.trap.rule.2                -          l2                Packets    269          269          0
                                                                        Bytes      37153        37153        0
lldp.ifp-0/1/2.trap.rule.0                 -          l2                Packets    269          269          0
                                                                        Bytes      37153        37153        0
lldp.ifp-0/1/2.trap.rule.1                 -          l2                Packets    269          269          0
                                                                        Bytes      37153        37153        0
lldp.ifp-0/1/2.trap.rule.2                 -          l2                Packets    269          269          0
                                                                        Bytes      37153        37153        0
```

#### Verifying Control Plane Policers

This command allows to view the policers created by the control-plane security feature.

**Syntax:**

**show qos policer** <options>

| Option | Description |
|---|---|
| - | Displays all policers, including those created by the control-plane security feature. |
| <policer-name> | Displays information about a specified policer. |
| counter | Displays all policer counters. |

Example 1: Display information of all policers created by the control-plane security feature.

```
supervisor@rtbrick: op> show qos policer
Policer: _DEFAULT_NAT_POLICER_50_MB
Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(Kbits)     PBS(Kbits)     Max CIR(Kbps)  Max PIR(Kbps)
  1        50000          50000          33000          33000          -              -
  2        -              -              -              -              -              -
  3        -              -              -              -              -              -
  4        -              -              -              -              -              -
Policer: _DEFAULT_POLICER_100_MB
Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(Kbits)     PBS(Kbits)     Max CIR(Kbps)  Max PIR(Kbps)
  1        100000         100000         33000          33000          -              -
  2        -              -              -              -              -              -
  3        -              -              -              -              -              -
  4        -              -              -              -              -              -
Policer: _DEFAULT_POLICER_10_MB
Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(Kbits)     PBS(Kbits)     Max CIR(Kbps)  Max PIR(Kbps)
  1        10000          10000          33000          33000          -              -
  2        -              -              -              -              -              -
  3        -              -              -              -              -              -
  4        -              -              -              -              -              -
Policer: _DEFAULT_POLICER_1_MB
Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(Kbits)     PBS(Kbits)     Max CIR(Kbps)  Max PIR(Kbps)
  1        1000           1000           33000          33000          -              -
  2        -              -              -              -              -              -
  3        -              -              -              -              -              -
  4        -              -              -              -              -              -
Policer: _DEFAULT_POLICER_20_MB
Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(Kbits)     PBS(Kbits)     Max CIR(Kbps)  Max PIR(Kbps)
  1        20000          20000          33000          33000          -              -
  2        -              -              -              -              -              -
  3        -              -              -              -              -              -
  4        -              -              -              -              -              -
Policer: _DEFAULT_POLICER_250_MB
Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(Kbits)     PBS(Kbits)     Max CIR(Kbps)  Max PIR(Kbps)
  1        250000         250000         33000          33000          -              -
  2        -              -              -              -              -              -
  3        -              -              -              -              -              -
  4        -              -              -              -              -              -
Policer: _DEFAULT_POLICER_500_MB
Type: two-rate-three-color, Levels: 1, Flags: -
```

```
  Level    CIR(Kbps)      PIR(Kbps)      CBS(Kbits)     PBS(Kbits)     Max CIR(Kbps)  Max PIR(Kbps)
  1        500000         500000         33000          33000          -              -
  2        -              -              -              -              -              -
  3        -              -              -              -              -              -
  4        -              -              -              -              -              -
Policer: _DEFAULT_POLICER_50_MB
Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(Kbits)     PBS(Kbits)     Max CIR(Kbps)  Max PIR(Kbps)
  1        50000          50000          33000          33000          -              -
  2        -              -              -              -              -              -
  3        -              -              -              -              -              -
  4        -              -              -              -              -              -
Policer: _DEFAULT_POLICER_5_MB
Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(Kbits)     PBS(Kbits)     Max CIR(Kbps)  Max PIR(Kbps)
  1        5000           5000           33000          33000          -              -
  2        -              -              -              -              -              -
  3        -              -              -              -              -              -
  4        -              -              -              -              -              -
Policer: _DEFAULT_POLICER_ARP_GLOBAL_10MB
Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(Kbits)     PBS(Kbits)     Max CIR(Kbps)  Max PIR(Kbps)
  1        10000          10000          33000          33000          -              -
  2        -              -              -              -              -              -
  3        -              -              -              -              -              -
  4        -              -              -              -              -              -
Policer: _DEFAULT_POLICER_BGP_GLOBAL_100MB
Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(Kbits)     PBS(Kbits)     Max CIR(Kbps)  Max PIR(Kbps)
  1        100000         100000         10000          10000          -              -
  2        -              -              -              -              -              -
  3        -              -              -              -              -              -
  4        -              -              -              -              -              -
Policer: _DEFAULT_POLICER_HOSTPATH_GLOBAL_50MB
Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(Kbits)     PBS(Kbits)     Max CIR(Kbps)  Max PIR(Kbps)
  1        50000          50000          5000           5000           -              -
  2        -              -              -              -              -              -
  3        -              -              -              -              -              -
  4        -              -              -              -              -              -
Policer: _DEFAULT_POLICER_IPoE_GLOBAL_50MB
Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(Kbits)     PBS(Kbits)     Max CIR(Kbps)  Max PIR(Kbps)
  1        50000          50000          5000           5000           -              -
  2        -              -              -              -              -              -
  3        -              -              -              -              -              -
  4        -              -              -              -              -              -
Policer: _DEFAULT_POLICER_ISIS_GLOBAL_50MB
Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(Kbits)     PBS(Kbits)     Max CIR(Kbps)  Max PIR(Kbps)
  1        50000          50000          5000           5000           -              -
  2        -              -              -              -              -              -
  3        -              -              -              -              -              -
  4        -              -              -              -              -              -
Policer: _DEFAULT_POLICER_LACP_GLOBAL_20MB
Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(Kbits)     PBS(Kbits)     Max CIR(Kbps)  Max PIR(Kbps)
```

```
  1        20000          20000          33000          33000          -              -
  2        -              -              -              -              -              -
  3        -              -              -              -              -              -
  4        -              -              -              -              -              -
Policer: _DEFAULT_POLICER_LDP_GLOBAL_50MB
Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(Kbits)     PBS(Kbits)     Max CIR(Kbps)  Max PIR(Kbps)
  1        50000          50000          5000           5000           -              -
  2        -              -              -              -              -              -
  3        -              -              -              -              -              -
  4        -              -              -              -              -              -
Policer: _DEFAULT_POLICER_LLDP_GLOBAL_10MB
Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(Kbits)     PBS(Kbits)     Max CIR(Kbps)  Max PIR(Kbps)
  1        10000          10000          1000           1000           -              -
  2        -              -              -              -              -              -
  3        -              -              -              -              -              -
  4        -              -              -              -              -              -
Policer: _DEFAULT_POLICER_OAM_GLOBAL_20MB
Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(Kbits)     PBS(Kbits)     Max CIR(Kbps)  Max PIR(Kbps)
  1        20000          20000          33000          33000          -              -
  2        -              -              -              -              -              -
  3        -              -              -              -              -              -
  4        -              -              -              -              -              -
Policer: _DEFAULT_POLICER_OSPF_GLOBAL_50MB
Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(Kbits)     PBS(Kbits)     Max CIR(Kbps)  Max PIR(Kbps)
  1        50000          50000          5000           5000           -              -
  2        -              -              -              -              -              -
  3        -              -              -              -              -              -
  4        -              -              -              -              -              -
Policer: _DEFAULT_POLICER_PPPOE_GLOBAL_50MB
Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(Kbits)     PBS(Kbits)     Max CIR(Kbps)  Max PIR(Kbps)
  1        50000          50000          5000           5000           -              -
  2        -              -              -              -              -              -
  3        -              -              -              -              -              -
  4        -              -              -              -              -              -
Policer: _DEFAULT_POLICER_RADIUS_GLOBAL_20MB
Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(Kbits)     PBS(Kbits)     Max CIR(Kbps)  Max PIR(Kbps)
  1        20000          20000          33000          33000          -              -
  2        -              -              -              -              -              -
  3        -              -              -              -              -              -
  4        -              -              -              -              -              -
Policer: _DEFAULT_POLICER_TRAP_20_MB
Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(Kbits)     PBS(Kbits)     Max CIR(Kbps)  Max PIR(Kbps)
  1        20000          20000          33000          33000          -              -
  2        -              -              -              -              -              -
  3        -              -              -              -              -              -
  4        -              -              -              -              -              -
Policer: _DEFAULT_POLICER_TRAP_TTL1
Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(Kbits)     PBS(Kbits)     Max CIR(Kbps)  Max PIR(Kbps)
  1        10000          10000          33000          33000          -              -
```

```
  2        -             -             -             -             -             -
  3        -             -             -             -             -             -
  4        -             -             -             -             -             -
```

Example 2: Display information of a specific policer

```
supervisor@rtbrick: op> show qos policer _DEFAULT_POLICER_50_MB
Policer: _DEFAULT_POLICER_50_MB
Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(Kbits)     PBS(Kbits)     Max CIR(Kbps)  Max PIR(Kbps)
  1        50000          50000          33000          33000          -              -
  2        -              -              -              -              -              -
  3        -              -              -              -              -              -
  4        -              -              -              -              -              -
```

Example 3: Display policer counters

```
supervisor@rtbrick: op> show qos policer counter
Interface                        Level  Units      Total            Received         Dropped
global_lldp                      1      Packets    55               55               0
                                        Bytes      7087             7087             0
arp_acl_untag                    1      Packets    1318             1318             0
                                        Bytes      84352            84352            0
arp_acl_double_tag               1      Packets    1318             1318             0
                                        Bytes      84352            84352            0
arp_acl_single_tag               1      Packets    1318             1318             0
                                        Bytes      84352            84352            0
ipv6_ll_prefix_acl               1      Packets    175              175              0
                                        Bytes      17656            17656            0
arp_acl_s_single_tag             1      Packets    1318             1318             0
                                        Bytes      84352            84352            0
arp_acl_sc_double_tag            1      Packets    1318             1318             0
                                        Bytes      84352            84352            0
ipv6_mcast_ff01_prefix_acl       1      Packets    175              175              0
                                        Bytes      17656            17656            0
ipv6_mcast_ff02_prefix_acl       1      Packets    175              175              0
                                        Bytes      17656            17656            0
lldp.ifp-0/1/0.trap.rule.0       1      Packets    55               55               0
                                        Bytes      7087             7087             0
lldp.ifp-0/1/0.trap.rule.1       1      Packets    55               55               0
                                        Bytes      7087             7087             0
lldp.ifp-0/1/0.trap.rule.2       1      Packets    55               55               0
                                        Bytes      7087             7087             0
lldp.ifp-0/1/2.trap.rule.0       1      Packets    55               55               0
                                        Bytes      7087             7087             0
lldp.ifp-0/1/2.trap.rule.1       1      Packets    55               55               0
                                        Bytes      7087             7087             0
lldp.ifp-0/1/2.trap.rule.2       1      Packets    55               55               0
                                        Bytes      7087             7087             0
lldp.ifp-0/1/3.trap.rule.0       1      Packets    55               55               0
                                        Bytes      7087             7087             0
lldp.ifp-0/1/3.trap.rule.1       1      Packets    55               55               0
```

```
                                        Bytes      7087             7087             0
lldp.ifp-0/1/3.trap.rule.2        1     Packets    55               55               0
                                        Bytes      7087             7087             0
lldp.ifp-0/1/5.trap.rule.0        1     Packets    55               55               0
                                        Bytes      7087             7087             0
lldp.ifp-0/1/5.trap.rule.1        1     Packets    55               55               0
                                        Bytes      7087             7087             0
lldp.ifp-0/1/5.trap.rule.2        1     Packets    55               55               0
                                        Bytes      7087             7087             0
lldp.ifp-0/1/6.trap.rule.0        1     Packets    55               55               0
                                        Bytes      7087             7087             0
lldp.ifp-0/1/6.trap.rule.1        1     Packets    55               55               0
                                        Bytes      7087             7087             0
lldp.ifp-0/1/6.trap.rule.2        1     Packets    55               55               0
                                        Bytes      7087             7087             0
lldp.ifp-0/1/7.trap.rule.0        1     Packets    55               55               0
                                        Bytes      7087             7087             0
lldp.ifp-0/1/7.trap.rule.1        1     Packets    55               55               0
                                        Bytes      7087             7087             0
lldp.ifp-0/1/7.trap.rule.2        1     Packets    55               55               0
                                        Bytes      7087             7087             0
lldp.ifp-0/1/8.trap.rule.0        1     Packets    55               55               0
                                        Bytes      7087             7087             0
lldp.ifp-0/1/8.trap.rule.1        1     Packets    55               55               0
                                        Bytes      7087             7087             0
lldp.ifp-0/1/8.trap.rule.2        1     Packets    55               55               0
                                        Bytes      7087             7087             0
lldp.ifp-0/1/10.trap.rule.0       1     Packets    55               55               0
                                        Bytes      7087             7087             0
lldp.ifp-0/1/10.trap.rule.1       1     Packets    55               55               0
                                        Bytes      7087             7087             0
lldp.ifp-0/1/10.trap.rule.2       1     Packets    55               55               0
                                        Bytes      7087             7087             0
lldp.ifp-0/1/12.trap.rule.0       1     Packets    55               55               0
                                        Bytes      7087             7087             0
lldp.ifp-0/1/12.trap.rule.1       1     Packets    55               55               0
                                        Bytes      7087             7087             0
lldp.ifp-0/1/12.trap.rule.2       1     Packets    55               55               0
                                        Bytes      7087             7087             0
lldp.ifp-0/1/14.trap.rule.0       1     Packets    55               55               0
                                        Bytes      7087             7087             0
lldp.ifp-0/1/14.trap.rule.1       1     Packets    55               55               0
                                        Bytes      7087             7087             0
lldp.ifp-0/1/14.trap.rule.2       1     Packets    55               55               0
                                        Bytes      7087             7087             0
lldp.ifp-0/1/16.trap.rule.0       1     Packets    55               55               0
                                        Bytes      7087             7087             0
lldp.ifp-0/1/16.trap.rule.1       1     Packets    55               55               0
                                        Bytes      7087             7087             0
lldp.ifp-0/1/16.trap.rule.2       1     Packets    55               55               0
                                        Bytes      7087             7087             0
lldp.ifp-0/1/22.trap.rule.0       1     Packets    55               55               0
                                        Bytes      7087             7087             0
lldp.ifp-0/1/22.trap.rule.1       1     Packets    55               55               0
```

```
                                        Bytes      7087            7087            0
lldp.ifp-0/1/22.trap.rule.2       1     Packets    55              55              0
                                        Bytes      7087            7087            0
lldp.ifp-0/1/24.trap.rule.0       1     Packets    23              23              0
                                        Bytes      3642            3642            0
lldp.ifp-0/1/24.trap.rule.1       1     Packets    23              23              0
                                        Bytes      3642            3642            0
lldp.ifp-0/1/24.trap.rule.2       1     Packets    23              23              0
                                        Bytes      3642            3642            0
lldp.ifp-0/1/26.trap.rule.0       1     Packets    55              55              0
                                        Bytes      7087            7087            0
lldp.ifp-0/1/26.trap.rule.1       1     Packets    55              55              0
                                        Bytes      7087            7087            0
lldp.ifp-0/1/26.trap.rule.2       1     Packets    55              55              0
                                        Bytes      7087            7087            0
<...>
```

This view displays multiple rules, such as the four PPPoE rules, all referencing the same policer. This shared reference explains why they exhibit identical counter values, as they point to a single policer instance.

Example 4: Display trap statistics

```
supervisor@rtbrick: op> show trap statistics
Trap Name                          Statistics ID   Units     Forwarded   Dropped
_RPF_STRICT_FAIL_TRAP              1               Packets   0           0
                                                   Bytes     0           0
_MTU_VIOLATED_TRAP                 2               Packets   0           0
                                                   Bytes     0           0
_DEFAULT_POLICER_TRAP_TTL1         528             Packets   0           0
                                                   Bytes     0           0
_MPLS_UNKNOWN_LABEL_TRAP           6               Packets   0           0
                                                   Bytes     0           0
_CPU_TRAP                          596             Packets   82          0
                                                   Bytes     7356        0
```

# 9.3. Local User Management

## 9.3.1. Local User Management Overview

Local User Management enables you to create, manage, and secure Linux local users and groups through the RBFS configuration using the RBFS CLI. This enables you to manage users and groups in the following environments:

- In the RBFS container only for virtual platform
- In the RBFS container and on the RtBrick Host OS for hardware platforms

Local User Management provides standalone authentication for console access, which is

independent of external AAA servers such as TACACS+. For details about configuring TACACS+ for RBFS, refer to the Securing the Management Plane User Guide.

#### Supported Platforms

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

### 9.3.2. Local user management Configuration

RBFS allows you to create privileges that are configurable for user-defined and pre-defined roles. RBFS supports a combination of permit and deny regular expressions and a configurable default privilege to support both blacklisting and whitelisting of users. If both permit and deny command regular expressions match, the allow regular expression takes precedence.

#### Creating Roles

To create a role, you need to configure the following:

- Configure role-based access control (RBAC) privilege for the role
- Configure the command privilege for the role

> It is important to have secure management enabled when granting any type of privilege, whether it is RBAC or command-based, to any user. Without secure management, privileges do not work.

##### Configuring the RBAC Privilege

You need to configure the RBAC privilege for both table and object.

**set system authorization global role** <name> **rbac-permission** ( **object** | **table** ) <resource> <permission-type>

#### Command arguments

| | |
|---|---|
| <name> | Authorization role name |
| <resource> | Represents resources in the RBFS (table/object) |

| | |
|---|---|
| <permission-type> | Permissions to create, read and delete. The following are the supported RBAC permission types:<br>-/-/-<br>-/-/delete<br>-/read/-<br>-/read/delete<br>create/-/-<br>create/-/delete<br>create/read/-<br>create/read/delete |

**Configuring the Command Privilege**

**set system authorization global role** <name> **cmd-permission** ( **allow-cmds** <allow-cmds> | **deny-cmds** <deny-cmds> )

| | |
|---|---|
| `<role>` | Authorization role name |
| `<allow-cmds>` | List of allow commands regular expression |
| `<deny-cmds>` | List of deny commands regular expression |

- If you configure a privilege for any of the pre-defined roles (supervisor, operator, reader), then it replaces the default privilege.
- If you delete the configured privilege for a pre-defined role, then it will revert to the default privilege of the role.
- Priority of privilege rules is as follows: explicit deny, explicit permit, default privilege.

The example below shows the new role named "support" which has RBAC permission to read any table and objects. Also, the user is denied everything except the allowed commands (ping, set, show, traceroute, and watch-mode).

```
{
  "ietf-restconf:data": {
    "rtbrick-config:system": {
      "authorization": {
        "global": {
          "role": [
            {
              "name": "support",
              "rbac-permission": [
                {
                  "permission": "-/read/-",
                  "resource-type": "object",
                  "resource": ".*"
                },
```

```
                {
                  "permission": "-/read/-",
                  "resource-type": "table",
                  "resource": ".*"
                }
              ],
              "cmd-permission": {
                "allow-cmds": [
                  "ping .*",
                  "set .*",
                  "show .*",
                  "traceroute .*",
                  "watch .*"
                  ],
                "deny-cmds": ".*"
              }
            }
          ]
        }
      }
    }
  }
}
```

**Linux pre-configured users, roles and privileges**

| User Name | Role Name | Default Privileges |
|---|---|---|
| supervisor | supervisor | Allow all actions. |
| operator | operator | All commands will be allowed except those that match any of the regular expressions below.<br>`"reboot.*",`<br>`"clear core-dump.*",`<br>`"request software.*"` |

| User Name | Role Name | Default Privileges |
|---|---|---|
| reader | reader | All commands will be denied except those that match any of the regular expressions below. expressions.<br>`"color.*",`<br>`"collect.*",`<br>`"date.*",`<br>`"exit.*",`<br>`"history.*",`<br>`"paging.*",`<br>`"ping.*",`<br>`"show.*",`<br>`"traceroute.*",`<br>`"watch-mode.*"` |

### Creating New Users

The new users created through local user management will always have a primary group with the same name and ID of the created user. The new user's ID will be allocated within the range of 3000 and 3999.

> You cannot use usernames such as `root`, `wheel`, `admin`, `sudo` or any of the SMP Linux pre-configured users and groups such as `supervisor`, `operator`, `reader`. Also, a username cannot start with "rtbrick_". If a Linux user with the same username already exists but has an ID outside of the 3000-3999 range then the user creation through the RBFS configuration will fail.

To create a new user, enter the following command:

**set system user** <username>

### Command arguments

| | |
|---|---|
| `<username>` | Name of the local user |

### Assigning Roles to Users

A "role" is an RBFS RBAC construct and it is mapped to a Linux group. The list of user roles from the RBFS configuration becomes the list of additional Linux groups that the Linux user belongs to. You can create new users and assign "roles" to the new users. The `supervisor`, `operator`, and `reader` are the pre-defined and pre-configured roles both in Linux and RBFS.

When a user is configured in RBFS under "system users", RBFS/confd validates that the list of user roles only contain roles that are pre-defined or that are configured under "system authorization".

> Do not create role names that start with "rtbrick_". In addition, "root", "wheel",

> "admin", and "sudo" are not acceptable role names.

To assign a role to a new user, enter the following command:

**set system user** <username> **role** <role>

### Command arguments

| | |
|---|---|
| `<username>` | Name of the user |
| `<role>` | Role of the user (not the primary role) |

Example: Assigning Roles to Users

```
{
  "ietf-restconf:data": {
    "rtbrick-config:system": {
      "user": [
        {
          "username": "bob",
          "role": [
            "operator"
            ],
          "shell": "/usr/local/bin/cli",
          "password-hashed-text":
"$6$uTE4OYn0iRq.Vppe$JBVMQ5DZHfuCuUP5yTnfl9IJsRLQAXqTLlLMKRO8bCz9WDlB2ele8puwMrT4/QDF2
nNOcoHtqYqFljly4B.Vu0"
        }
      ]
    }
  }
}
```

### Configuring Authentication for a New User

There are two ways in which you can create a password for a new user in RBFS:

1. Creating hashed password
2. Creating plain-text password

#### Creating Hashed Password

You can verify the integrity of your password using hashed passwords. When a user is present in the configuration but a "password hashed text" is not present, the password authentication is considered disabled for that specific user.

Three predefined roles include `supervisor`, `operator`, and `reader`. `supervisor` is the default password for the role supervisor. For the 'operator' and 'reader' roles, there is no default password.

Use the `system users supervisor` command to disable the default supervisor password.

You can also disable password authentication for any of the predefined users such as `supervisor`, `operator` and `reader` by adding a "system users supervisor" configuration section without any "password hashed text" and thus disabling password authentication for the supervisor user.

SSH public keys can be configured even if "password hashed text" is not present.

To create a password hashed text and authenticate the new user, perform the following steps:

1. Generate hash password on any Linux server.
   `mkpasswd --method=SHA-512`
2. Configure authentication using a password hashed text and an SSH public key.
   **set system user** <username> **password-hashed-text** <password-hashed-text>
   **set system user** <username> **ssh-pub-key** <ssh-pub-key>

#### Changing an Existing Hashed Password for Supervisor

You can change an existing hashed password for a 'supervisor' user after logging in to the system.

**Syntax:**

> **set system user** <username> **password-hashed-text** <password-hashed-text>
> **set system user** <username> **ssh-pub-key** <ssh-pub-key>

Example Command:

```
set system user supervisor password-hashed-text
$6$HIPXmd0uiLD0RtcT$3V7YxFJWR3b6NEGGd41RCT0TseWgdKLFAl6RecvDOqIUnaCHMt0zo0ZZR4/1
```

**Command arguments**

| | |
|---|---|
| `<username>` | Name of the user. |
| `<password-hashed-text>` | Password string. |
| `<ssh-pub-key>` | public keys of a user. You can specify multiple ssh-pub-keys. |

3. Log in using username and password hashed text.

- It is possible to change shell, password-hash and ssh-pub-keys for supervisor, reader, and operator roles.
- The password string provided as part of the RBFS configuration needs to be a compatible password hashed text as defined by the shadow manual page: https://manpages.debian.org/buster/passwd/shadow.5.en.html and by the crypt https://manpages.debian.org/buster/manpages-dev/crypt.3.en.html.

### Example


```
{
  "ietf-restconf:data": {
    "rtbrick-config:system": {
      "user": [
        {
          "username": "bob",
          "shell": "/usr/local/bin/cli",
          "password-hashed-text":
"$5$L2DaOYYuddhBV$9RA5MX9RQzLC9fIKJzbnoFBb88w9rkSXl7GVrVJ9PY7",
          "ssh-pub-key": [
            "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQCBAAABAQCubg5sdDycPN5EViNkV6w7rfp2GAfKWuInfaL3xOXyvSNpsmaHILYmgr
LUU0GKQH9gauPUJpDcvvYaMt0ZBuTbWHVMUc4cvhgbNDkTB2bG2cTZ5QzbicyXff3BlDWQThVp2LtVBiW2tf7J
TTa9SnL4Lnm+CQcXsQ0rxqy2S6bJpsRYlFMyQl/hZ4QEWE153dw0HGvcG8mjfnPN4wvCc/omfD3ljxx+Gf4oFS
0davX6pdphUKLvgL33VVG5xaK71imv2l3897LIJZaHxy7FbB+CjSYT6QNq1XksX8omrbRjiP3enEQi/bANtzTN
nGDnIm1KHf3xuKpoKw+B5fhDZogx"
            ]
        }
      ]
    }
  }
}
```


#### Creating Plain-Text Password

To configure a password with plain text password for a new user, use the following command.

### Syntax:

**set system user** <username> **password-plain-text** <password-plain-text>

### Command arguments

| | |
|---|---|
| <username> | Name of the user |
| <password-plain-text> | Specifies the plain-text password |

### Example for Configuring Plain Text Password

```
set system user bob password-plain-text bob123
set system user bob shell /usr/local/bin/cli
set system user bob role operator
```

#### Changing an Existing Plain-Text Password for Supervisor

You can change an existing plain text password for the 'supervisor' user after logging into the system. Use the following commands to change an existing password.

**Syntax:**

**set system user** <username> **password-plain-text** <password-plain-text>

Sample Commands:

```
set system user supervisor shell /bin/bash
set system user supervisor password-plain-text rtbrick
```

#### Viewing Configuration of Plain Text Password


```
{
  "ietf-restconf:data": {
    "rtbrick-config:system": {
      "user": [
        {
          "username": "bob",
          "role": [
            "operator"
            ],
          "shell": "/usr/local/bin/cli",
          "password-hashed-text":
"$6$uTE4OYn0iRq.Vppe$JBVMQ5DZHfuCuUP5yTnfl9IJsRLQAXqTLlLMKRO8bCz9WDlB2ele8puwMrT4/QDF2
nNOcoHtqYqFljly4B.Vu0"
        }
      ]
    }
  }
}
```


#### Setting the User Shell

RBFS validates that the shell is one of the following 3 valid options:

- /usr/sbin/nologin
- /bin/bash
- /usr/local/bin/cli

To configure user shell, enter the following command:

**set system user** <username> **shell** <shell>

#### Command arguments

| | |
|---|---|
| `<username>` | Name of the user |
| `<shell>` | Name of the shell |

#### Example

```
root@rtbrick: cfg> set system user smith shell /usr/local/bin/cli
```

#### Specifying the Display Name for User Names

The display name allows you to specify a preferred name so that you can easily identify the user. You can change your display name by entering the following command:

**set system user** <username> **display-name** <display_name>

#### Command arguments

| | |
|---|---|
| `<username>` | Name of the user |
| `<display_name>` | Display name to easily identify the user |

#### Example

```
set system user smith display-name primeuser
```

#### Enabling or disabling CLI access

You can control a user's access to the CLI. By default, users will have access to the CLI.

**set system user** <username> **no-cli-access** [ **true** | **false** ]

#### Command arguments

| | |
|---|---|
| `<username>` | Name of the user |
| `<true \| false>` | When the no-cli-access is set to `true`, the user's access to the CLI is disabled. When the no-cli-access is set to `false`, the user will be able access the CLI. |

#### Example

```
set system user smith no-cli-access false
```

#### Collecting Debug Information

You can collect the debug information using the following command:

```
set system user <username> role debug_info
```

#### Configuring sudo Without Password

You can configure local system users to log in via passwords or using SSH keys. From a security perspective, it is desirable to allow authentication with SSH keys only. RBFS provides a configuration knob to disable the requirement for a 'sudo' password so that local users can authenticate with SSH keys only. This knob is configurable only if the user or one of its roles is `supervisor`.

You can enter the following command to enable or disable the 'sudo' password. By default, this is set to false which ensures that the supervisor must provide a password when using sudo.

**set system user** <user> **no-sudo-password** < **true** | **false** >

#### Command arguments

| | |
|---|---|
| `<username>` | Name of the user |
| `<true \| false>` | When the `no-sudo-password` is set to `true`, it indicates that a 'sudo' password is not required. When it is set to `false`, it indicates that the supervisor must provide a password when using sudo. |

Example Configuration:

```
{
    "rtbrick-config:user": [
      {
        "username": "smith",
        "role": "supervisor",
        "shell": "/bin/bash",
        "ssh-pub-key": "ssh-rsa AAAAB3Nza<...>",
        "no-sudo-password": "true"
      }
    ]
 }
```

If 'no-sudo-password' is set, you can log in with your SSH key.

**Configuring Fail2Ban**

The failed SSH login attempts from one user over an SSH jump host affect other users from the same jump host. You can configure Fail2Ban, which enables you to whitelist some IP addresses by creating separate jails for each user connecting through the jump host. In this way, failed login attempts by one user will not affect another user.

The following Fail2Ban command allows you to configure a list of IP addresses to include those IP addresses in the whitelist. You can specify multiple IP addresses that you want to exclude from the ban. Fail2Ban is applicable to both the RtBrick Host and the Linux container.

**Syntax**:

**set system platform-management fail2ban ignore-ip** <ignore-ip>

**Command arguments**

| | |
|---|---|
| `ignore-ip <ignore-ip>` | Specify the IP addresses in CIDR notation which are to be whitelisted. |

Example commands for configuration:

```
supervisor@rtbrick>LEAF01: cfg> show config set
set system
set system platform-management fail2ban ignore-ip 10.1.1.1/32
set system platform-management fail2ban ignore-ip 10.1.1.2/32
set system platform-management fail2ban ignore-ip 10.2.2.0/24
```

Example Configuration:

```
supervisor@rtbrick>LEAF01: cfg> show config
{
  "ietf-restconf:data": {
    "rtbrick-config:system": {
      "platform-management": {
        "fail2ban": {
          "ignore-ip": [
            "10.1.1.1/32",
            "10.1.1.2/32",
            "10.2.2.0/24"
            ]
        }
      }
    }
  }
```

```
}
```

### Enabling or Disabling Password-based Authentication

The "set system authentication ssh" CLI allows modifying the SSH configurations on the device. The `password-authentication` option disables the password-based authentication for user login. By default, password-based authentication is enabled. Additionally, two more SSH authentication parameters can be configured: `client-alive-count-max` and `client-alive-interval`.

> After disabling password-based authentication, the device will not be accessible through password-based login; hence, you must configure a public key for the user before disabling this. If you have disabled password authentication by mistake and forgot to configure the public key, you can only recover the device using RESTCONF by configuring the password authentication to “true.”

**Syntax:**

**set system authentication ssh** <attribute> <value>

| Attribute | Description |
|---|---|
| `client-alive-count-max` | Defines the maximum number of unanswered keepalive messages before terminating the connection. |
| `client-alive-interval` | Sets the interval (in seconds) for sending keepalive messages to the client to prevent idle disconnections. |
| `password-authentication` | Enables or disables password-based authentication, set to 'false' to enforce key-based authentication for enhanced security. |

The example below shows disabling password-based authentication and logging in using SSH key.

```
supervisor@rtbrick: cfg> set system authentication ssh password-authentication false
supervisor@rtbrick: cfg> commit
supervisor@rtbrick: cfg> set system user supervisor
supervisor@rtbrick: cfg> set system user supervisor shell /bin/bash
supervisor@rtbrick: cfg> set system user supervisor ssh-pub-key "ssh-ed25519
ABCDEFC1lZDI1NTE5AAcRsYHrrTesQr04UuSrtefN/ndezZM7+8mBC"
supervisor@rtbrick: cfg> commit
supervisor@rtbrick: cfg>
```

**Configuration Example:**

```
{
    "ietf-restconf:data": {
        "rtbrick-config:system": {
            "authentication": {
                "ssh": {
                    "password-authentication": "false"
                }
```

```
            },
            "user": [
                {
                    "username": "supervisor",
                    "shell": "/bin/bash",
                    "ssh-pub-key": [
                        "ssh-ed25519 ABCDEFC1lZDI1NTE5AAcRsYHrrTesQr04UuSrtefN/ndezZM7+8mBC"
                    ]
                }
            ]
        }
    }
}
```

The example below shows how to log into the device after disabling password-based authentication.

```
supervisor@rtb:~$ ssh supervisor@bl1-pod1
Last login: Wed Feb 26 07:28:06 2025 from 192.168.1.135
supervisor@rtb:~$ ssh supervisor@bl1-pod1
Last login: Wed Feb 26 07:28:06 2025 from 192.168.1.135

+-------------------------------------------------------------------+
|                                                                   |
    System rtbrick running on ufi15.q2c.u15.r5.nbg.rtbrick.net:
        Date:       Thu Nov 20 06:21:58 AM UTC 2025
        Uptime:     up 1 hour, 2 minutes

    Image metadata:
        UUID:       0aef95f4-3d2a-4bdc-90aa-784231ad9e5d
        Version:    25.4.1-candidate.3
        Role:       multiservice-edge
        Platform:   q2c
        Model:      s9600-72xc
        Format:     lxd
        Build date: 2025-11-19 08:52:54 UTC
        Based on:   Debian GNU/Linux 12 (bookworm)

    RBFS License:
        No license found
|                                                                   |
+-------------------------------------------------------------------+
```

The example below show how to re-enable the password-based authentication and log in to the device.

```
supervisor@rtbrick: cfg> set system authentication ssh password-authentication true
supervisor@rtbrick: cfg> commit
supervisor@rtbrick: cfg> delete system user supervisor
```

```
supervisor@rtbrick: cfg> commit
supervisor@rtbrick: cfg>
```

**Configuration Example:**

```
{
    "ietf-restconf:data": {
        "rtbrick-config:system": {
            "authentication": {
                "ssh": {
                    "password-authentication": "true"
                }
            }
        }
    }
}
```

The example below shows how to log into the device after enabling password-based authentication.

```
ssh supervisor@ufi15.q2c.u15.r5.nbg.rtbrick.net
supervisor@ufi15.q2c.u15.r5.nbg.rtbrick.net's password:
+-------------------------------------------------------------------+
|                                                                   |
    RBFS container rtbrick running on ufi15.q2c.u15.r5.nbg.rtbrick.net:
        Date:       Thu Nov 20 06:21:58 AM UTC 2025
        Uptime:     up 1 hour, 2 minutes

    Image metadata:
        UUID:       0aef95f4-3d2a-4bdc-90aa-784231ad9e5d
        Version:    25.4.1-candidate.3
        Role:       multiservice-edge
        Platform:   q2c
        Model:      s9600-72xc
        Format:     lxd
        Build date: 2025-11-19 08:52:54 UTC
        Based on:   Debian GNU/Linux 12 (bookworm)

    RBFS License:
        No license found
|                                                                   |
+-------------------------------------------------------------------+
```

### 9.3.3. LUM Operational Commands

The show commands to display information about local and TACAS+ users and their sessions. You can use the command and options to view the summary of all users, information for a specific user and to view user sessions details.

### LUM Show Commands

#### User Details

**Syntax:**

**show user** <option>

| Option | Description |
|---|---|
| session <session-id> | Displays information about user session ID specified. |

Example: The output shows the currently logged-in users along with their roles, authentication methods, source IP addresses, login time and session details.

```
supervisor@rtbrick: op> show user
Flags: * - Current Session
Username            Role                Shell     SSH-Key   Authenticator  Remote Host        Login Time
Session ID
supervisor          supervisor          bash      no        Local          10.200.137.164     2026-06-05 12:54:47
4*
```

Example: The output shows the detailed information about a specific user session, including the login method, role, shell, service, and connection details such as host, session duration, and idle time.

```
supervisor@rtbrick: op> show user session 4
Session Information:
    Name            : supervisor
    Session ID      : 4
    TTY             : pts/5
    Display Name    : supervisor
    Host            : 10.200.137.164
    PID             : 659
    Public Key      : no
    Login Method    : Local
    Roles           : supervisor,debug_info,rtbrick_confd_sock_access
    Shell           : bash
    Service         : ssh
    Login Time      : Fri Jun 05 12:54:47 GMT +0000 2026
    Session Duration: 24s
    Idle Timestamp  : Fri Jun 05 12:54:47 GMT +0000 2026
    Idle Duration   : 24s
```

## 9.4. sFlow

### 9.4.1. RBFS sFlow Technology

**sFlow Overview**

sFlow is a packet-sampling protocol for monitoring network traffic. sFlow enables continuous, real-time monitoring of network traffic across all interfaces simultaneously. It works by collecting samples of data packets and transmitting these samples in UDP datagrams to a central station called collector.

The sampling process is directly handled by the networking processing ASIC, which ensures accuracy in packet sampling and also significantly reduces the utilization of control-plane resources such as CPU, memory.

sFlow provides insights into network traffic that enables the network operator to promptly identify performance issues, security threats, or unusual traffic patterns. It allows real-time network monitoring across thousands of router ports simultaneously in a large scale environment.

**Packet-Based Sampling:**

Packet-based sampling is a technique for capturing a subset of network traffic by selecting one packet from a predefined number of packets. It generates a representative sample of the overall traffic without capturing every packet, which would be resource-intensive.

- RBFS sFlow mechanism captures only the packet headers passing through the incoming and outgoing physical interfaces.
- The sFlow header for Q2A-based platforms (Edgecore CSR440 and UfiSpace S9510-28DC) includes the 'In-ifindex' and 'Out-ifindex' fields.

**Benefits of sFlow Technology**

sFlow offers the following benefits:

**Detect Network Issues and Manage Real-time Congestion**: sFlow provides visibility into traffic patterns and enables the detection of anomalies such as misconfigurations, packet loss, and latency and it can track unauthorized activities. It can monitor traffic flow and bandwidth usage in real-time and provides data to manage network congestion.

For example, if an RBFS device suddenly experiences high latency or packet loss, sFlow provides information to pinpoint the source of the issue, such as a faulty interface or a congested port.

**Types of Applications and Usage Patterns**: By analyzing packet headers, sFlow can identify different types of network traffic such as web browsing, peer-to-peer (P2P), and DNS queries. This helps you to understand the types of applications and the amount of bandwidth that they consume. It allows to track changes in resource usage.

For example, if a network experiences an increase in P2P traffic, sFlow can provide the cause and the users or applications that are involved. With this data, network administrators can take appropriate actions.

**Billing and Charge-Back**: sFlow can provide resource usage data on per-application or per-user

basis, which is useful for enterprise customers that require billing or internal charge-back. Enterprises can utilize this data to accurately bill customers or to allocate costs to internal departments based on the actual network consumption.

**Route Profiling and Peering Optimization**: sFlow captures detailed routing information and traffic distribution that helps to analyze traffic patterns across different routes and optimize peering agreements. It helps to refine routing policies and peering relationships.

For example, service providers can use sFlow data to determine which peering partners receive most of the traffic. This information allows you to modify their peering policies to improve performance.

**Trends and Capacity Planning**: sFlow can provide data on traffic trends and bandwidth usage that helps you to increase capacity.

For example, you can see a particular link is nearing its bandwidth capacity by analyzing sFlow data. With this data, you can plan an upgrade.

### Understanding RBFS sFlow Implementation

sFlow works by capturing sample network packets and collecting data periodically to give a representative view of the overall traffic. It does not capture every packet, instead, it samples a subset of packets. For example, it captures one packet in every N packets (where N is configurable) on a physical interface.

#### sFlow Components

An RBFS sFlow system comprises an sFlow agent embedded in a device and a remote sFlow collector.

### sFlow Agent

The sFlow agent gathers data by sampling packets from the ports which are configured for sFlow. It then encapsulates this data into UDP datagram and sends the samples to the collector. The sFlow collector analyzes these samples to give you data about traffic patterns, issues, and performance of the network.

sFlow Agent is a key component and performs the following tasks:

1. Sampling to collect packet samples based on the defined number of packets
2. Process the collected packets
3. Export the UDP datagram to the sFlow collector

For sending the sampled packets, you can configure a destination port. The default destination port is 6343.

### sFlow Collector

The collector receives the sampled data from the sFlow agents. It aggregates, processes, and stores the data for analysis. The collector can analyze and interpret metrics such as packet types,

bandwidth usage, IP addresses, protocol information, and so on.

**Flow Sampling and Number of Samples**

With the packet sampling, an sFlow agent samples packets on a designated interface based on a sampling rate that is set.

Each sFlow packet carries sampled data, and the number of samples determines the amount of data included in a single packet. In RBFS, the number of samples in every sFlow packet by default is 1. It means each sFlow packet contains one sampled record.

**Guidelines for sFlow Settings**

- To successfully transport sFlow data to the collector, make sure the routing table includes a route that enables communication with the collector's configured IPv4 address. Specifically, there should be a route entry for the collector's IPv4 address within the instance where the sFlow collector is configured. This ensures that the device knows the correct path to reach the collector's IP address within the specified routing instance, allowing sFlow data transit to the collector.
- The agent ID must be identical across all collectors configurations within a device. This ensures the source of sampled data is recognized as the same across multiple collectors.
- It is required to enable sFlow on each interface individually by attaching a traffic-sampling profile. This profile defines the collector IP address, sampling direction and so on.
- You can configure a maximum number of four different collectors to receive sampled traffic.

> When the interface is attached to a traffic-sampling profile, attaching a mirror to the same interface may prevent sampling from working.

**Supported Platforms**

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

## 9.4.2. sFlow Configuration

**Configuration Hierarchy**

The following diagram illustrates the sFlow configuration hierarchy. In this configuration hierarchy, traffic sampling collectors and traffic sampling profiles are set up under forwarding-options to enable packet sampling using the sFlow protocol. After defining the collectors and sampling profiles, the sampling profile can be attached to the physical interfaces that allows the device to sample traffic from that interfaces.

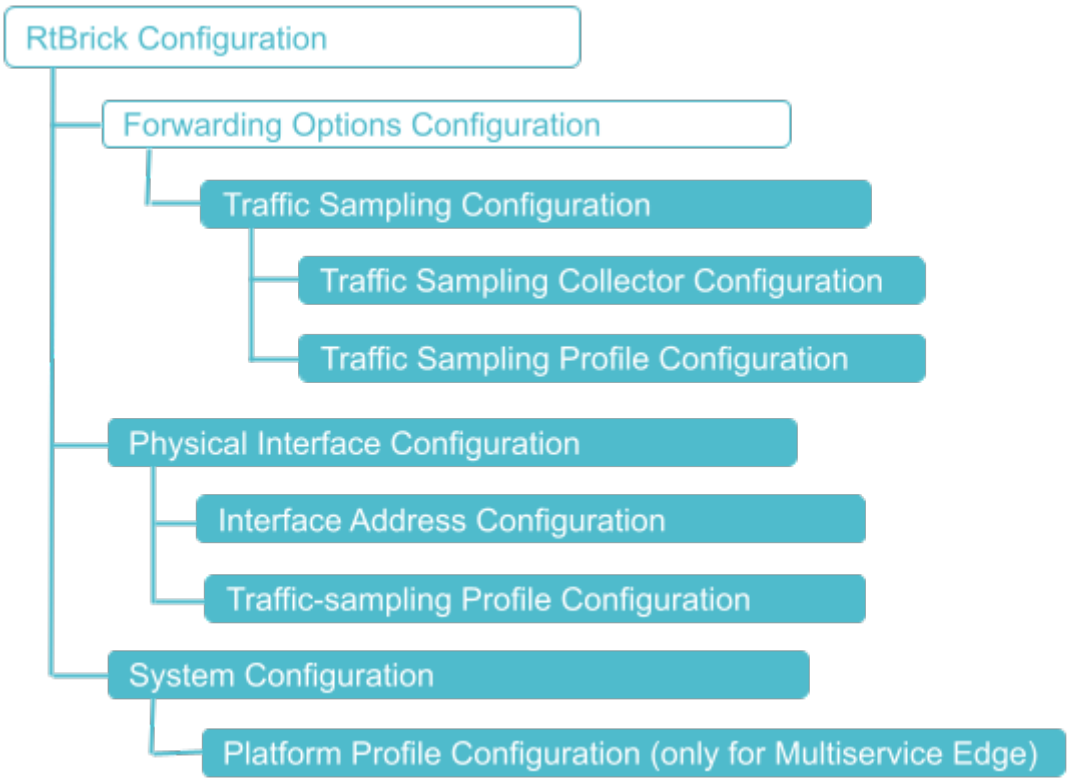



*Figure 157. sFlow Configuration Hierarchy*

**Configuration Syntax and Commands**

The following sections describe the sFlow configuration syntax and commands.

**Prerequisites:**

Before configuring sFlow, ensure that the appropriate platform profile is configured on the device. The platform profile determines the operating mode of the system and enables the required hardware resources for sFlow functionality.

You must configure "sFlow" parameter for the Platform Profile for on multiservice edge devices only. The Spine and Access-Leaf devices do not require this platform profile settings. You must reboot the device for the changes to take effect after completing the Platform Profile configuration.

For information about platform profile configuration, see Platform Profile Configuration.

**Syntax**

**set system platform profile** <profile-name>

| Attribute | Description |
|---|---|
| `profile_name` | Name of platform profile |
| `feature` | Feature name. |

Example commands:

This configuration defines a platform profile named 'nat_4q' that enables the sFlow feature.

```
set system platform profile nat_4q
set system platform profile nat_4q feature sFlow
```

```
supervisor@rtbrick.net: op> show config system platform
{
  "rtbrick-config:platform": {
    "profile": [
      {
        "profile_name": "nat_4q",
        "feature": [
          "sFlow"
          ]
      }
    ]
  }
}
```

#### sFlow Collector Configuration

You can configure a maximum number of four collectors to receive sampled traffic.

> The agent ID must be identical across all collectors. This ensures the source of sampled data is recognized as the same across multiple collectors.

You can create, modify, or delete collectors dynamically, while the device continues to operate. This gives you the flexibility to modify the configurations of the sFlow collectors without causing any downtime.

### Syntax

**set forwarding-options traffic-sampling collector** <collector-id> <attribute> <value>

Example commands:

```
set forwarding-options traffic-sampling collector 1
set forwarding-options traffic-sampling collector 1 instance default
set forwarding-options traffic-sampling collector 1 export-protocol sflow
set forwarding-options traffic-sampling collector 1 address ipv4 10.1.1.1
set forwarding-options traffic-sampling collector 1 sampling-rate 100
set forwarding-options traffic-sampling collector 1 destination-udp-port 6343
set forwarding-options traffic-sampling collector 1 agent-id 1.2.3.4
```

| Attribute | Description |
|---|---|
| `collector-id` | Identifier for each collector. |
| `agent-id` | Unique identifier for the device sending sFlow data. This ID must be the same across all collectors. |
| `collector-ip` | The IP address of the collector to which the sFlow UDP data gram is sent. |
| `instance` | The Instance of collector IP reachability. |
| `destination-udp-port` | The UDP port used for sending sFlow data to the collector (Default: 6343). |
| `sampling-rate` | Specifies the rate at which the packets are sampled. Range: 100 - 50,000. Example: When configured as 100, 1 packet out of 100 packets gets sampled. |
| `export-protocol` | The protocol for sending the captured sample packets to the collector (Protocol used is sFlow). |

Example: Collector configuration

The following configuration sets up sFlow collectors. Each collector is assigned with a unique ID and is configured with an IP address to receive the sampled traffic, along with parameters like sampling rate, number of samples, and the destination UDP port for sFlow data. The same agent ID (1.2.3.4) is used for all collectors. The sampled traffic is sent to the collector's IP address, 10.1.1.1, over UDP port 6343. All collectors operate within the 'default' instance.

Collector 1 is configured with a sampling rate of '1' that indicates every packet traversing the configured interface is sampled without skipping any packet. The number of samples specified is 3, which indicates that the individual packets are aggregated into groups of three samples before being exported to the collector. The sFlow agent, which is the device or system sending the traffic samples, is defined by the agent identifier '1.2.3.4'.

In this configuration, collectors 2, 3, and 4 sample every 10th packet but differ in the number of samples and destination ports. Agent ID 1.2.3.4 is configured for all collectors 2, 3, and 4.

```
{
    "rtbrick-config:forwarding-options": {
        "traffic-sampling": {
            "collector": [
                {
                    "collector-id": 1,
                    "instance": "default",
                    "export-protocol": "sflow",
                    "address": {
                        "ipv4": "10.1.1.1"
                    },
                    "sampling-rate": 100,
                    "destination-udp-port": 6343,
                    "agent-id": "1.2.3.4"
```

```
                },
                {
                    "collector-id": 2,
                    "instance": "default",
                    "export-protocol": "sflow",
                    "address": {
                        "ipv4": "20.1.1.1"
                    },
                    "sampling-rate": 100,
                    "destination-udp-port": 6343,
                    "agent-id": "1.2.3.4"
                },
                {
                    "collector-id": 3,
                    "instance": "default",
                    "export-protocol": "sflow",
                    "address": {
                        "ipv4": "30.1.1.1"
                    },
                    "sampling-rate": 100,
                    "destination-udp-port": 6343,
                    "agent-id": "1.2.3.4"
                },
                {
                    "collector-id": 4,
                    "instance": "default",
                    "export-protocol": "sflow",
                    "address": {
                        "ipv4": "40.1.1.1"
                    },
                    "sampling-rate": 100,
                    "destination-udp-port": 6343,
                    "agent-id": "1.2.3.4"
                }
            ]
        }
    }
}
```


#### Traffic-Sampling Profile Configuration

You can create a traffic sampling profile that can be attached to the physical interface .

### Syntax

**set forwarding-options traffic-sampling profile** <profile-name> **collector-id** <collector-id> **description** <description> **direction** [ingress\|egress\|both]

Example set of commands:

```
set forwarding-options traffic-sampling profile test
set forwarding-options traffic-sampling profile test collector 1
set forwarding-options traffic-sampling profile test direction ingress
set forwarding-options traffic-sampling profile test export-protocol sflow
```

| Attribute | Description |
|---|---|
| `profile-name` | Name of the sampling profile. |
| `collector-id` | The identifier of the collector to which the packet samples are sent. |
| `direction` | Defines whether traffic sampling is performed on ingress, egress, or both of the directions. |
| `description` | Description for the profile. |

Example: Traffic sampling profile configuration

This configuration defines two traffic sampling profiles that control how network traffic is sampled and exported to specific collectors using the sFlow protocol. Each profile focuses on capturing only incoming traffic on the monitored interfaces and exports this data to a designated collector for analysis.

The first profile, named "test", is configured to send the collected traffic data to Collector 1. This profile is designed to focus on incoming traffic.

The second profile, named "test1", has a similar configuration. It also monitors only incoming traffic, using the sFlow protocol to export the data. The traffic collected under this profile is directed to Collector 2.

```
{
    "rtbrick-config:forwarding-options": {
        "traffic-sampling": {
            "profile": [
                {
                    "profile-name": "test",
                    "collector": 1,
                    "direction": "ingress",
                },
                {
                    "profile-name": "test1",
                    "collector": 2,
                    "direction": "ingress",
                }
            ]
        }
    }
}
```

#### Attach Traffic-Sampling Profile with the Physical Interface

You can attach a traffic sampling profile to a physical interface using the following command.

### Syntax

**set interface** <interface-name> **traffic-sampling-profile** <profile-name>

Example Command:

```
set interface ifp-0/1/10 traffic-sampling-profile test1
```

| Attribute | Description |
|---|---|
| `interface-name` | Name of the physical interface. |
| `<profile-name>` | Name of the profile. |

Example: Traffic sampling profile is attached on multiple physical interfaces.

This configuration sets up a few physical interfaces with different speeds. Some of these interfaces have traffic sampling profiles configured. Two interfaces (`ifp-0/1/10` and `ifp-0/1/8`) are configured with traffic sampling profiles ('test1' and 'test', respectively).

```
[
    {
        "name": "ifp-0/1/10",
        "speed": "10G",
        "traffic-sampling-profile": "test1",
        "unit": [
            {
                "unit-id": 1,
                "address": {
                    "ipv4": [
                        {
                            "prefix4": "121.0.0.1/24"
                        }
                    ],
                    "ipv6": [
                        {
                            "prefix6": "fc66:1337:7331::1/64"
                        }
                    ]
                }
            },
            {
                "unit-id": 25,
                "vlan": 25,
                "address": {
```

```
                    "ipv4": [
                        {
                            "prefix4": "10.1.1.2/24"
                        }
                    ]
                }
            },
            {
                "unit-id": 26,
                "vlan": 26,
                "address": {
                    "ipv4": [
                        {
                            "prefix4": "20.1.1.2/24"
                        }
                    ]
                }
            }
        ]
    },
    {
        "name": "ifp-0/1/20",
        "speed": "10G"
    },
    {
        "name": "ifp-0/1/26",
        "speed": "1G"
    },
    {
        "name": "ifp-0/1/5",
        "speed": "1G"
    },
    {
        "name": "ifp-0/1/6",
        "speed": "1G"
    },
    {
        "name": "ifp-0/1/61",
        "speed": "10G"
    },
    {
        "name": "ifp-0/1/63",
        "speed": "10G"
    },
    {
        "name": "ifp-0/1/8",
        "speed": "10G",
        "traffic-sampling-profile": "test"
    },
    {
        "name": "ifp-0/1/9",
```

```
        "speed": "10G"
    },
    {
        "name": "lo-0/0/4",
        "unit": [
            {
                "unit-id": 1,
                "address": {
                    "ipv4": [
                        {
                            "prefix4": "1.2.3.4/24"
                        }
                    ]
                }
            }
        ]
    }
]
```

### 9.4.3. sFlow Operational Commands

#### sFlow Collector Information

The `show traffic-sampling sflow collector` command output provide details about the sFlow collectors configured on the device. It shows the information related to each collector, including its sampling rates, associated agent, destination IP address, and other configuration specifics.

**Syntax:**

**show traffic-sampling sflow profile** <profile-name> **collector** <collector-id>

| Option | Description |
|---|---|
| `<profile-name>` | Name of the profile. |
| `collector-id` | Identifier for the collector. |

Example: Summary Output for sFlow collectors

```
supervisor@rtbrick: op> show traffic-sampling sflow collector
Collector ID: 1
    Agent-ID: 198.51.100.21
    Address: 192.0.2.2
    Instance: default
    Sampling-rate: 100
    UDP-Destination-Port: 6343
```

Example: Summary Output for a specified sFlow collector

```
supervisor@rtbrick: op> show traffic-sampling sflow collector 1
Collector ID: 1
    Agent-ID: 198.51.100.21
    Address: 192.0.2.2
    Instance: default
    Sampling-rate: 100
    UDP-Destination-Port: 6343
```

#### Sampling Profiles

Displays the packet sampling profile related information.

**Syntax:**

**show traffic-sampling profile** <options>

| Option | Description |
|---|---|
| `<profile-name>` | Name of the profile. |

Example: Summary Output for traffic sampling profiles

```
supervisor@rtbrick: op> show traffic-sampling sflow profile
Profile name: test
    Collector Id: 1
    Direction: ingress
```

Example: Summary Output for traffic sampling for a specified profile

```
supervisor@rtbrick: op> show traffic-sampling sflow profile test
Profile name: test
    Collector Id: 1
    Direction: ingress
```

##### Display Traffic Sampling Interface

The `show traffic-sampling interface` command displays detailed traffic sampling configuration information for a given interface. It provides visibility into the sampling profile, collector ID, traffic direction, and export protocol.

**Syntax:**

**show traffic-sampling interface**
**show traffic-sampling interface** <interface-name>

The following table provide the command options and descriptions.

| Attribute | Description |
| --- | --- |
| `<interface-name>` | Name of the interface. |

The following example displays traffic sampling configuration details for all interfaces with traffic sampling enabled.

```
supervisor@rtbrick.net: cfg> show traffic-sampling interface
Interface name: ifp-0/0/0
  Profile name: test_1
  Collector Id: 1
  Direction: ingress
  Export protocol: sflow
Interface name: ifp-0/0/1
  Profile name: test_1
  Collector Id: 1
  Direction: ingress
  Export protocol: sflow
```

The following example displays traffic sampling configuration details for the specified interface.

```
supervisor@rtbrick.net: cfg> show traffic-sampling interface ifp-0/0/1
Interface name: ifp-0/0/1
  Profile name: test_1
  Collector Id: 1
  Direction: ingress
  Export protocol: sflow
```

# 10. Telemetry

Telemetry provides real-time monitoring, gathers historical data, and uses standard network management tools to provide clear, actionable insights across your network.

It offers complete visibility into the status, health, and performance of network devices. You can easily collect, monitor, store, and export data for troubleshooting. Telemetry enables you to make informed decisions.

Key Features:

- Monitor system health and resource utilization, including CPU, memory, processes, disk usage, hardware sensors, power supplies, fans, clock synchronization, and optical transceivers for proactive fault detection.
- Collect, store, and export time-series operational metrics for various network services with configurable metric collection and retention. Integrate with Prometheus and Grafana to export telemetry metrics for historical analysis, visualization, dashboarding, alerting, and capacity planning.
- Support hardware-assisted sFlow packet sampling and interface counter export to provide scalable traffic visibility, top-talker analysis, traffic engineering insights, and network utilization monitoring.
- Support SNMP-based monitoring of system, interface, hardware, routing, and protocol statistics using standard and enterprise MIBs for seamless integration with network management systems.

The following guides are available under the Telemetry section:

- TSDB
- Resource Monitoring
- SNMP
- sFlow

## 10.1. TSDB

### 10.1.1. Observability

RBFS runs Prometheus, an open-source monitoring tool with a built-in time-series database. It gathers metrics (data) from RBFS and its microservices (brick daemons). Prometheus uses a polling model and pulls metrics data.

RBFS includes a pre-packaged Prometheus software and it does not require a separate Prometheus installation. However, to begin collecting and storing data, you must enable Prometheus observability on an RBFS device. It remains inactive until a time-series has been configured for the first time.

Prometheus observability is an optional service in RBFS.

### Understanding Prometheus Observability in RBFS

Prometheus operates as a microservice in the RBFS container and it queries at regular intervals to pull metrics data directly from various RBFS brick daemons or processes. It collects metrics and stores them with the timestamp, which shows when the data was recorded. RBFS supports direct export of metrics to Prometheus.

By monitoring and analyzing these metrics, you can assess the health of your applications, detect any anomalies, and take necessary corrective actions. You can define custom metrics.

Prometheus pulls metrics periodically from all targeted processes at a scrape interval. A scrape interval indicates how frequently these metrics are collected.

At a high level, three key operations take place:

- Prometheus scrapes metrics from various applications and services at regular intervals.
- The collected data is stored in a time-series database for querying and analysis.
- Alert rules analyze the data and triggers alerts to notify about the critical events or anomalies.

### Example of Scenarios When or How Time-Series Metrics Data Help

The following are a few example scenarios of how and when the observability data help network administrators:

- Detecting Memory Leaks: A gradual increase in memory usage without a corresponding change in the PPPoE session count indicates a memory leak.
- Predicting Hardware Failures: A persistent five-minute chassis temperature readings signals an imminent hardware fault, prompting a replacement.
- Identifying Fabric Congestion: Consistent 80 percent utilization across all fabric interfaces suggests the need to add new fabric links.
- Detecting Weakening Optical Links: When a network device receives a high volume of incoming traffic and simultaneously shows a drop in optical receive power, suggests that the optical signal is weakening.

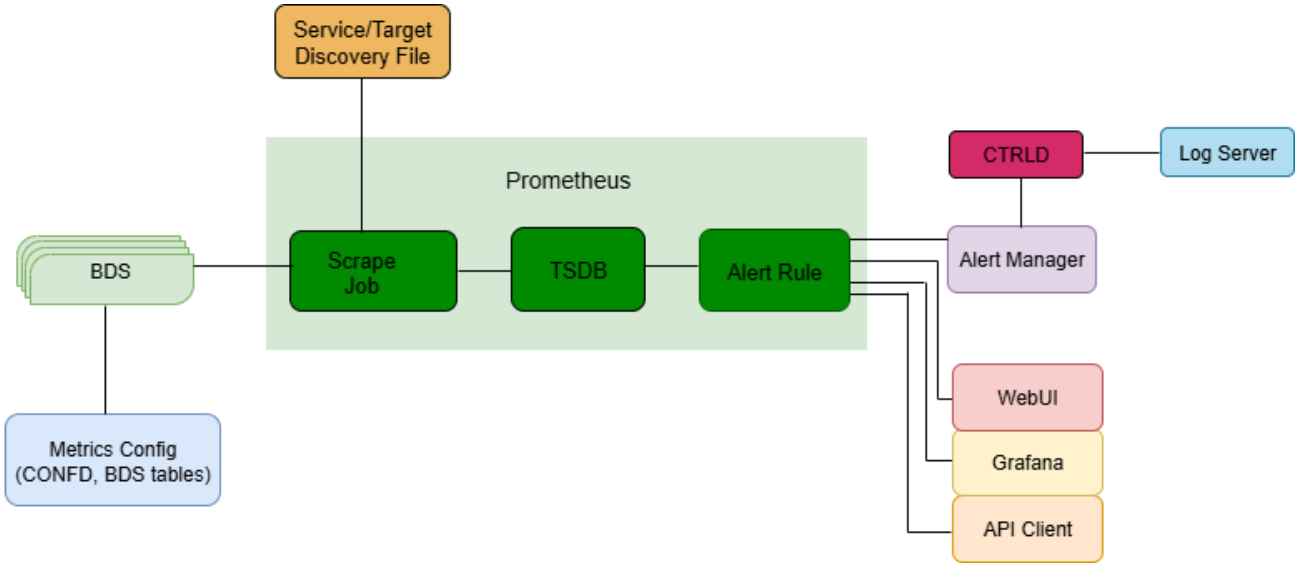



*Figure 158. Prometheus Monitoring in RBFS*

The communication uses the Prometheus exposition format, which operates through an HTTP pull from the metrics endpoint of the scrape target.

RBFS brick daemons communicate to the configuration table called `global.time-series.metric.config`, which lists the metrics a daemon is supposed to export. Only the BDS, which holds the data table for each daemon, responds with metrics. The data scraped by Prometheus is stored in its own local time-series database on the device.

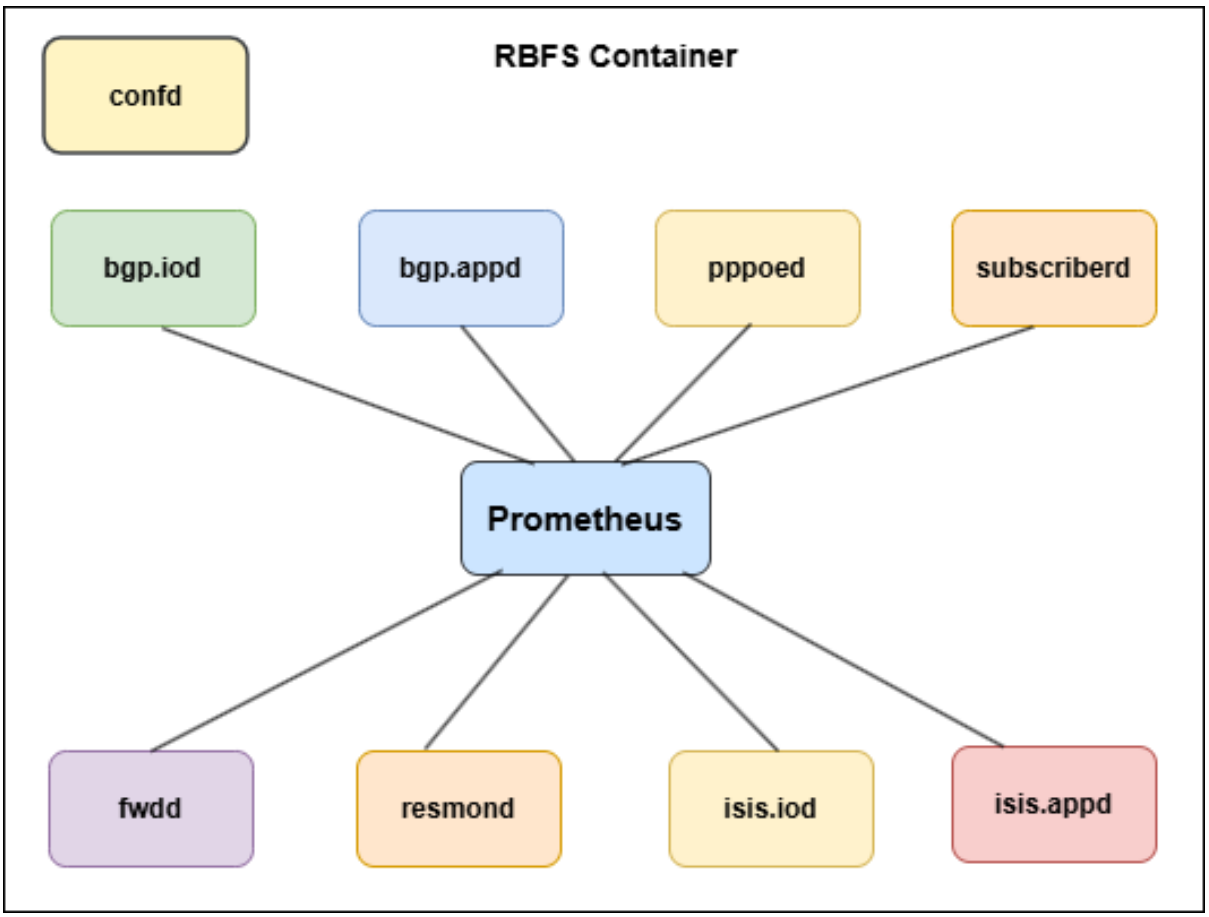



*Figure 159. Prometheus in RBFS with the different scrape target*

**Visualization**

RBFS supports Grafana, a visualization platform, to present metrics data visually. For more information, see Grafana.

#### Federated Deployment of Prometheus

In a federated Prometheus deployment model, multiple Prometheus instances can be deployed hierarchically in large-scale environment. A global Prometheus instance aggregates data from other Prometheus nodes enabling hierarchical data collection. This hierarchical structure helps manage large volumes of time-series data. It is useful for environments where a single Prometheus instance is inadequate.

Prometheus Federation allows you to consolidate metrics from multiple Prometheus instances into a single, centralized instance.

The following `prometheus.yml` file demonstrates the global Prometheus instance configured to scrape metrics from local Prometheus instances using the '/federate' endpoint:

```
global:
  scrape_interval: 60s

scrape_configs:
  - job_name: "federate"
    honor_labels: true
    metrics_path: '/federate'
    params:
      'match[]':
        - '{job=~"^bds_.+"}'
    scrape_interval: 15s
    file_sd_configs:
      - files:
          - ./bds.target.yml
        refresh_interval: 5m
```

The 'match[]' parameter indicates the global Prometheus instance to retrieve only time-series metrics associated with the BDS job. This ensures efficient data selection and eliminates the need to modify the configuration when new aggregation rules are introduced.

The Prometheus targets, which are the local Prometheus instances from which to pull data, are defined in the 'bds.target.yml' file:

```
- targets: ['198.51.100.125:19091']
  labels:
    __metrics_path__:
"/api/v1/rbfs/elements/rtbrick/services/prometheus/proxy/federate"
    box: 125_rtbrick
```

This configuration specifies the endpoint to scrape and assigns custom labels to the data, such as identifying the RBFS device.

For more information about Prometheus Federation, see Prometheus Federation.

#### Control Daemon

Control Daemon (CtrlD) is a key component that facilitates service discovery, state monitoring, and proxying for other services within a monitoring architecture. It interacts with Prometheus for metrics collection and Alertmanager for alert management.

CtrlD operates as a proxy and acts as a middle layer to handle requests without querying Prometheus directly. Both Prometheus and Alertmanager are registered with CtrlD, allowing it to be aware of their existence and track their operational state (whether they are up or down). Alertmanager manages alerts triggered by Prometheus based on predefined conditions. When an alert is triggered in Alertmanager, CtrlD forwards these alerts to external destinations, such as Grafana.

CtrlD can also be used to make Prometheus queries via HTTP. For example:

```
curl
'http://198.51.100.125:19091/api/v1/rbfs/elements/rtbrick/services/prometheus/proxy/ap
i/v1/query?query=up' | jq .
```

This command queries Prometheus to check the status of the services to see if they are up.

For federation, the following URL allows for sharing data between multiple Prometheus instances:

```
http://198.51.100.125:19091/api/v1/rbfs/elements/rtbrick/services/prometheus/proxy/fed
erate
```

**Down-sampling**:

RBFS supports down-sampling for Prometheus federated deployment. It helps to reduce the amount of stored data by averaging, aggregating, or sampling over a specified period of time. The mechanism involves grouping the data over time periods and calculating things based on averages, sums, or maximums.

### Metrics and Metric Profiles

#### Metrics

Metrics are data points that are recorded with a timestamp to create a metric time-series. The metric as such is the current value. Storing the metric values with a timestamp forms a metric time-series which can then be scanned for anomalies or used for trend analysis. Metrics are often accompanied by labels, which provide additional context about the data. For example, a CPU usage metric can include labels that indicate the instance or application that it belongs to.

**Index Metrics and Object Metrics**:

Index Metric and Object Metrics are two different ways of storing and querying time-series data. They differ in how they are used to store and retrieve time-series data. Index-metrics refer to

metrics stored within the index of Prometheus.

Object-metrics are the actual data points, stored within the Prometheus. They represent the measured values over time.

### Metric Filters

Filters allow you to include or exclude specific BDS objects based on the objects attributes. The filtering is only applicable on 'bds-type object-metric', it is not supported on 'index-metric'.

For more information, see Metrics.

#### Attributes

Attributes represent the numeric fields (columns) in BDS tables that are monitored and exported as time-series metrics. These are the actual data points such as packet counters, byte counters and so on. Attributes are key elements because they determine the specific information to be collected. The attributes specify which state will be exported.

> This attribute must be either a numeric type or a type that can be converted to numeric.

#### Labels

Labels (Attribute Labels) provide additional context and metadata about metrics. They are key-value pairs attached to metrics. Labels help differentiate between different instances of the same attribute. For example, when Prometheus monitors multiple applications simultaneously, each application may have a `system_memory_used_bytes` metric. Labels enable Prometheus to store multiple metrics from various sources under a single metric name.

Moreover, attribute labels are useful for filtering, grouping, and analyzing metrics across different entities. They are necessary for uniquely identifying metric streams, especially when the same attribute appears in multiple rows of a table.

For more information, see Metrics Labels.

#### Filters

Filters (Attribute Filters) are used to narrow down the objects of a table should be considered for metric generation. In some tables, there may be hundreds of entries, but you may only track metrics for a few specific rows. Filters let you define conditions (match rules) that rows must meet to be included in metric generation.

#### Metric Profiles

In RBFS, you can create metric profiles that allow you to group multiple metrics which share the same characteristics, such as metrics from the same application or instances. Instead of configuring each metric individually, you can define the scrape interval and the append-timestamp flag commonly for all associated metrics in a metric profile.

If you do not explicitly assign a metric to a specific metric profile, it will be by default assigned to

the default metric profile. This default profile is set to scrape metrics every 5 seconds. If you do not create or use any custom metric profile, all metrics will automatically fall under this default profile. Also, if you do not define any custom metric profile, Prometheus scrapes all daemons that expose metrics in every 5 seconds.

### Alert and Alert Profiles

#### Alert

You can configure Prometheus to generate alerts based on defined rules, which are then managed by the Alert Manager.

Alerts are triggered based on rules that evaluate time-series data stored in the Prometheus. When the specified conditions are met, the alert manager notifies CtrlD which forwards the notification to the configured GELF or syslog servers.

For more information, see Alerting and Alertmanager.

#### Alert Profile

An alert profile allows you to group multiple alerts into a single profile by applying the same characteristics to them. The alert profile also controls the repeat interval, which specifies how often a notification will be repeated if the underlying alert condition remains unresolved.

#### Tracking Invalid Time-series Configurations

RBFS provides the ability to track invalid time-series configurations applied to RBFS. This functionality helps to debug Prometheus time-series that are not in the expected state (for example, missing labels) due to configuration settings that cause runtime issues. This functionality is enabled by default. You can view the list of misconfigurations using the show command: `show time-series metric <metric-name> rtbrick_invalid_metric_config`.

#### Monitoring Daemon Status

The built-in Prometheus 'up' metrics must not be used to monitor the brick daemons. Prometheus scrapes only metrics from daemons that actually expose metric data. RBFS allows you to monitor the state of the daemons whether they are operational or not. You can enable the functionality. For configuration information, see section Monitoring Daemon Status.

#### Prometheus Self-Monitoring

With the self-monitoring functionality, Prometheus can monitor its own performance and health. This functionality gathers metrics related to Prometheus' own operations that include query performance and resource usage.

#### Append Timestamp

RBFS allows you to enable Append Timestamp on both metric or metric profile level. When the append timestamp flag is set, the first metric generated after an object is updated will include the actual timestamp of the update. This accurately reflects when the change took place. For all subsequent metrics for the same object (since it has not been updated again) the current scrape

time will be used.

#### Storage

On average, Prometheus uses only about 1 to 2 bytes per sample. Therefore, to plan the capacity of a Prometheus server, you can use the following rough formula:

```
needed_disk_space = retention_time_seconds * ingested_samples_per_second *
bytes_per_sample
```

The disk space for the single binaries:

```
-rwxr-xr-x 1 root root 27M Sep 2 22:51 alertmanager +
-rwxr-xr-x 1 root root 81M Sep 2 22:51 prometheus +
-rwxr-xr-x 1 root root 49M Sep 3 19:55 promtool
```

Use the 'promtool' for testing configurations before setting them in Prometheus. For information, see promtool.

For more information about storage, see Storage.

### 10.1.2. Time-series Configurations in RBFS

#### Configuration Overview

RBFS does not require a separate Prometheus software installation as it includes a pre-packaged Prometheus software. The time-series configuration defines how time-series data is collected, stored, and transported to external systems. The configuration allows RBFS to stream time-series data from its microservices and processes to the integrated Prometheus collector.

It allows you to define metric profiles, create individual metrics, and associate those metrics with specific metric profile.

RBFS time-series metric configurations at a high level:

- **Metric Profile Configuration**: Define common settings that include scrape interval and append timestamp for a group of metrics.
- **Metric Configuration**: Configure the parameters of the metrics which are to be collected and exported. It includes the BDS table that contains the operational state to be converted into a time-series. You can define attributes, filters, and labels. They help in tagging, querying, filtering, and grouping the data in the time-series database.
  - **Attribute Configuration** (Mandatory): Define the BDS attributes that will be exported.
  - **Define Filters** (Optional): Filters the BDS objects which are going to be exported based on defined criteria. This is required when the BDS table contains more than one object.
  - **Define Labels** (Optional): It identifies and clarifies time services coming from different BDS objects by labeling them with metadata. Allows for querying, grouping, and filtering in time-

series data.

- **Alert Configuration**: Inform about specific conditions in the system based on the captured metrics.
- **Alert Profile Configuration**: Group multiple alerts into a single profile by applying notification rules to alerts with similar characteristic.
- **Enabling Data History Retention**: Option for debugging. If enabled, BDS logs the BDS queries that were executed to retrieve the telemetry data from BDS.
- **Enabling Prometheus Self Monitoring**: Prometheus can monitor its own health state.
- **Monitoring Daemon Status**: Monitor the status of system daemons (processes), specifically to track whether they are operational or not.

> In previous releases, there was a built-in Prometheus 'UP' metric to monitor the status of the daemon and determine whether it was up or down. However, this built-in functionality is no longer available. You can now define your own custom metric to achieve the same monitoring capabilities.

**Configuration Hierarchy**

The diagram illustrates the Prometheus configuration hierarchy.

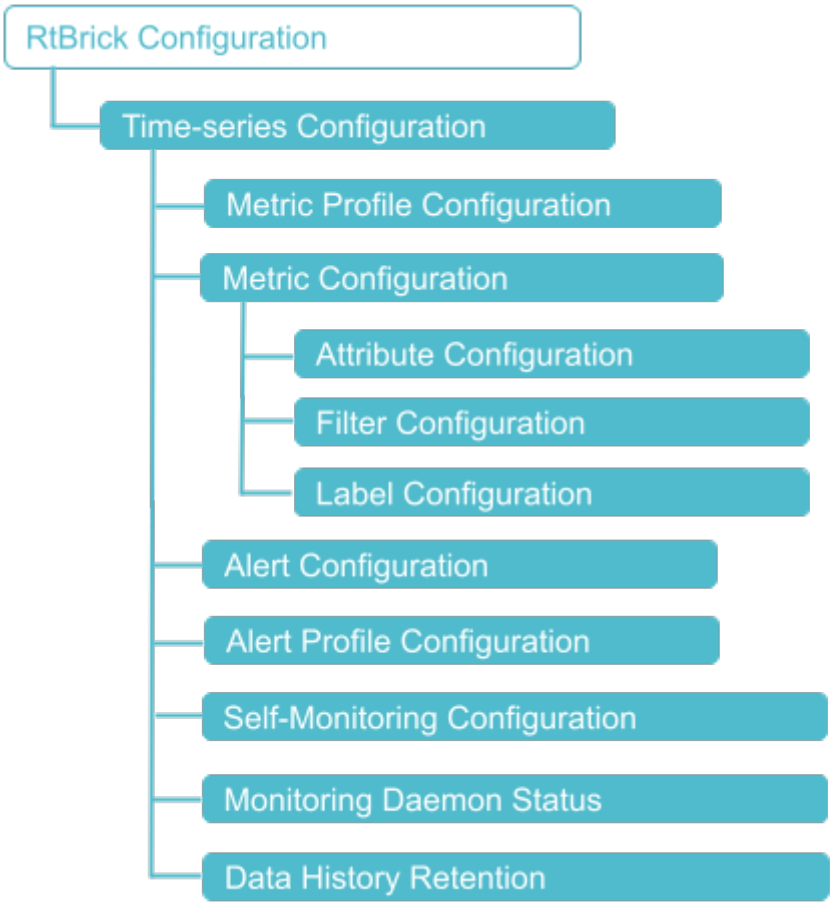



*Figure 160. TSDB Configuration Hierarchy*

### Configuration Syntax and Commands

The following sections describe the time-series configuration syntax and commands.

#### Guidelines for Naming

Metric and Metric Profile names must not start with "rtbrick," as this name is reserved for internal use. Also, RBFS has built-in default metric profile; therefore, the custom metric profile names must not be "default".

#### Metric Configuration

Metric configuration is used to configure the operational state which is to be exported.

- RBFS time-series configuration may differ from device to device; however, the exported metrics should remain consistent because the characteristics of these metrics will be the same across all platforms. For example, while the names of the chassis thermal sensors may vary from platform to platform, the underlying metric will consistently be the chassis temperature metric.
- Every metric must be attached to a metric profile. If you do not explicitly associate a metric with a profile, it will automatically be assigned to the default metric profile.

**Syntax:**

**set time-series metric** <name> <option> <value>

| Attribute | Description |
|---|---|
| `metric_name` | Name of the metric. See Naming Conventions. |
| `append-timestamp <true \| false>` | Set the append-timestamp to true for exporting the metric values with timestamp. By default, the value is 'false'. The timestamp is the epoch represented in milliseconds, and its value corresponds to the creation time of the current metrics in RBFS. |
| `attribute` | Specifies the name of the attribute in the target table which is to be scraped and exported. |
| `bds-type <object-metric \| index-metric>` | Specifies the type of metric, that is scraped and exported. There are two types, object-metric and index-metric.<br>• object-metric: Object metrics are scraped from the BDS objects.<br>• index-metric: Index metrics are scraped from the table index. |

| Attribute | Description |
| --- | --- |
| `metric_type` | • gauge: A gauge is a metric that represents a single numerical value that can fluctuate up and down. Gauges are commonly used to measure values such as temperature or current memory usage, as well as "counts" that can increase or decrease, like the number of concurrent requests.<br>• counter: A cumulative metric that represents a single value that only increases over time or resets to zero upon a restart. It is ideal for tracking quantities such as the number of requests served, tasks completed, or errors encountered.<br>Do not use a counter to expose values that can decrease. Counters are strictly for values that only increase or reset on restart. Use gauge for metrics that may increase or decrease. |
| `description` | Description for the metric. Allowed range of 128 characters. |
| `filter <match-attribute-name>` | Specifies the matching attribute name for the filter. |
| `include-subscribed-tables <true \| false>` | Specifies whether the configuration needs to be applied on a subscribed tables as well. Default: false. |
| `index-name` | Specifies the index-name if the 'bds_metric_type' is index-metric. This is applicable for index-metric only. |
| `prometheus-type <counter \| gauge>` | Configures the Prometheus metric type. Currently, the supported Prometheus metric types are 'counter' and 'gauge'. |
| `table-name` | Specifies the target table from which the data is scraped and exported (can be a regular expression). |

API To access the RESTCONF API that corresponds to this CLI, click here.

Example:

This configuration is used to export fan speed telemetry data from the sensor table for only chassis fans. It defines a time-series metric named 'chassis_fan_speed_rpm' for telemetry that captures chassis fan speed (in RPM) from a sensor table called `global.chassis_0.resource.sensor.fan` (which contains sensor data, including fans).

The 'bds-type' is defined as 'object-metric', which means the metric is based on object-like data entries in the table. The 'prometheus-type' is specified as 'gauge', that is for numeric values that can go up or down. The description for the metric is added as 'Chassis fan speed in rpm'.

A filter is applied to the 'resource_name' column, with the 'match-attribute-value' set to 'Chassis'. The match-type is defined as 'regular-expression'. This filter determines which rows in the source table (`global.chassis_0.resource.sensor.fan`) will be included when generating this metric. It specifically matches rows where the value of 'resource_name' satisfies the regular expression for 'Chassis'.

Also, a second filter is applied to the 'resource_type', which only includes rows where the value is exactly 'fan'. It ensures an exact string match. This makes sure that only fan-related resource entries are considered and filtered out irrelevant sensors.

```
set time-series metric chassis_fan_speed_rpm
set time-series metric chassis_fan_speed_rpm table-name
global.chassis_0.resource.sensor.fan
set time-series metric chassis_fan_speed_rpm include-subscribed-tables false
set time-series metric chassis_fan_speed_rpm bds-type object-metric
set time-series metric chassis_fan_speed_rpm prometheus-type gauge
set time-series metric chassis_fan_speed_rpm description "Chassis fan speed in rpm"
set time-series metric chassis_fan_speed_rpm filter resource_name
set time-series metric chassis_fan_speed_rpm filter resource_name match-attribute-
value Chassis
set time-series metric chassis_fan_speed_rpm filter resource_name match-type regular-
expression
set time-series metric chassis_fan_speed_rpm filter resource_type
set time-series metric chassis_fan_speed_rpm filter resource_type match-attribute-
value fan
set time-series metric chassis_fan_speed_rpm filter resource_type match-type exact
set time-series metric chassis_fan_speed_rpm attribute rpm
set time-series metric chassis_fan_speed_rpm attribute rpm label fan
set time-series metric chassis_fan_speed_rpm attribute rpm label fan label-value
resource_name
set time-series metric chassis_fan_speed_rpm attribute rpm label fan label-type
dynamic
```

Example:

```
supervisor@rtbrick.net: cfg> show config time-series metric chassis_fan_speed_rpm
{
  "rtbrick-config:metric": [
    {
      "name": "chassis_fan_speed_rpm",
      "table-name": "global.chassis_0.resource.sensor",
      "bds-type": "object-metric",
      "prometheus-type": "gauge",
      "description": "Chassis fan speed in rpm",
      "filter": [
        {
          "match-attribute-name": "resource_name",
          "match-attribute-value": "CHASSIS.*",
          "match-type": "regular-expression",
          "match-negate": "false"
        },
        {
          "match-attribute-name": "resource_type",
          "match-attribute-value": "fan",
          "match-type": "exact",
```

```
          "match-negate": "false"
        }
      ],
      "attribute": [
        {
          "attribute-name": "rpm",
          "label": [
            {
              "label-key": "fan",
              "label-value": "resource_name",
              "label-type": "dynamic"
            },
            {
              "label-key": "status",
              "label-value": "status_code",
              "label-type": "dynamic"
            }
          ]
        }
      ]
    }
  ]
}
```

**Example for Index Metric Configuration**

The following is an example for index metric configuration.

This configuration defines a time-series metric called 'local_bds_statistics.' It retrieves data from the source table named 'local.bds.statistics.' The 'bds-type' is classified as an index-metric, indicating that this metric is based on an index. The Prometheus type is specified as 'counter,' which means this is a counter metric. The 'index-name' is set to 'immutable-index.'

The attribute 'active-entry-count' is defined to specify a particular attribute from the source table that will be included in the metric. Labels are assigned to indicate the attributes. This configuration enables the export of the 'active-entry-count' from the 'local.bds.statistics' table into Prometheus as a counter-type metric, tagged with a static label and indexed using the immutable index.

```
set time-series metric local_bds_statistics
set time-series metric local_bds_statistics table-name local.bds.statistics
set time-series metric local_bds_statistics bds-type index-metric
set time-series metric local_bds_statistics prometheus-type counter
set time-series metric local_bds_statistics index-name immutable-index
set time-series metric local_bds_statistics description "This metric is exported from
local.bds.statistics by all BDs"
set time-series metric local_bds_statistics attribute active-entry-count
set time-series metric local_bds_statistics attribute active-entry-count label
active_entry_count
set time-series metric local_bds_statistics attribute active-entry-count label
active_entry_count label-value active_entry_count
```

```
set time-series metric local_bds_statistics attribute active-entry-count label
active_entry_count label-type static
```

```
supervisor@rtbrick.net: cfg> show config time-series metric local_bds_statistics
{
  "rtbrick-config:metric": [
    {
      "name": "local_bds_statistics",
      "table-name": "local.bds.statistics",
      "bds-type": "index-metric",
      "prometheus-type": "counter",
      "index-name": "immutable-index",
      "description": "This metric is exported from local.bds.statistics by all BDs",
      "attribute": [
        {
          "attribute-name": "active-entry-count",
          "label": [
            {
              "label-key": "active_entry_count",
              "label-value": "active_entry_count",
              "label-type": "static"
            }
          ]
        }
      ]
    }
  ]
}
```

**Example Metric Configuration for Monitoring Daemon Status**

This configuration defines a metric named `process_status`, which extracts data from a specific internal system table called `global.process.status`. The BDS type is defined as object-metric, indicating that the metric is derived from an object table. In the configuration, the metric type is specified as a gauge. It indicates increase or decrease, for example 1 for UP and 0 for DOWN.

The description provided is "testing UP metrics." The main attribute is defined as `operational_status`, and its values will be exported as part of the metric from the table. Additionally, the configuration introduces a label called `administrative_state` for the metric. The value for this label will be sourced from the `admin_status` column in the same row of the table. It is marked as dynamic, meaning the label value can vary at runtime for each row of data.

```
set time-series metric process_status
set time-series metric process_status table-name global.process.status
set time-series metric process_status bds-type object-metric
set time-series metric process_status prometheus-type gauge
set time-series metric process_status description "monitor daemon status metrics"
set time-series metric process_status attribute operational_status
```

```
set time-series metric process_status attribute operational_status label
administrative_state
set time-series metric process_status attribute operational_status label
administrative_state label-value admin_status
set time-series metric process_status attribute operational_status label
administrative_state label-type dynamic
set time-series metric process_status attribute operational_status label bd
set time-series metric process_status attribute operational_status label bd label-
value process
set time-series metric process_status attribute operational_status label bd label-type
dynamic
```

Example:

```
{
  "rtbrick-config:metric": [
    {
      "name": "process_status",
      "table-name": "global.process.status",
      "bds-type": "object-metric",
      "prometheus-type": "gauge",
      "description": "monitor daemon status metrics",
      "attribute": [
        {
          "attribute-name": "operational_status",
          "label": [
            {
              "label-key": "administrative_state",
              "label-value": "admin_status",
              "label-type": "dynamic"
            },
            {
              "label-key": "bd",
              "label-value": "process",
              "label-type": "dynamic"
            }
          ]
        }
      ]
    }
  ]
}
```

To access the RESTCONF API that corresponds to this CLI, click here.

#### Metric Attributes

Attributes are used for extracting specific values from the table data.

The number of labels is limited to 50 labels per metric.

#### Metric Filters

Filters allow you to include or exclude specific BDS objects based on the objects attributes.

The list of attribute filters determines which table rows are included in metric generation. All filters in this list must match to generate the metric, so the list functions as an implicit AND.

#### Metric Labels

Labels are key-value metadata on metrics. For example, it adds related labels such as CPU ID, interface name, and so on.

Each unique combination of key-value label pairs creates a new time series, which can significantly increase the amount of data stored. Avoid using labels to represent dimensions with high cardinality, such as user IDs, email addresses, or other sets of values that can vary widely.

To access the RESTCONF API that corresponds to this CLI, click here.

#### Tracking Invalid Time-series Configurations

RBFS offers functionality to track invalid time-series configurations that cause runtime issues. For example, when Prometheus reports duplicate time-series data due to wrong labels.

You can view the list of misconfiguration using the show command:

```
show time-series metric rtbrick_invalid_metric_config
```

There are six error types for invalid configurations in RBFS:

| Error Type | Description |
|---|---|
| `Error Type 1: invalid_filter` | This error occurs due to a invalid filter name. If the configured filter attribute is not present in the table, this error is shown. If this error occurs, the filter does not have any effect, and there may be many objects get exported. |
| `Error Type 2: invalid_attribute` | This error occurs due to a invalid attribute name. If the configured attribute is not present in the table, then this error is shown. If this error occurs, the expected metric does not appear as the configured attribute does not exist. |

| Error Type | Description |
| --- | --- |
| `Error Type 3: invalid_attribute_type` | This error occurs due to a invalid type of attribute. If the configured attribute is present in the table and if the datatype is wrong, then this error is shown. if this error occurs, the expected metric does not show up, because the attribute does not contain a numeric value or its value cannot be converted into a numeric value. |
| `Error Type 4: invalid_label` | This error occurs due to a invalid key name. If the configured label attribute is not present in the table, then this error is shown. If this error occurs, Prometheus may process duplicate time-series entries due to missing labels. Filtering time-series does not work. |
| `Error Type 5: invalid_label_attribute_type` | This error occurs due to a invalid label attribute type. If the configured label attribute is present in the table and if the datatype is wrong, this error is shown. the configured label attribute is not present in the table, then this error is shown. If this error occurs, Prometheus may process duplicate time-series entries due to missing labels. Filtering time-series does not work. |
| `Error Type 6: invalid_filter_attribute_type` | This error occurs due to a invalid filter attribute type. If the configured filer attribute is present in the table and if the datatype is wrong, this error is shown. |

The following sections provide examples for various error types with invalid configurations:

#### Error Type 1: Invalid Filter

In this configuration, the filter name 'invalid_filter' is used as a key instead of specifying a valid attribute. So, this is not a valid configuration.

```
set time-series metric bgp_default_peer_state filter invalid_filter
set time-series metric bgp_default_peer_state filter invalid_filter match-attribute-
value Active
set time-series metric bgp_default_peer_state filter invalid_filter match-type exact
```

```
supervisor@rtbrick.net: cfg> show config time-series metric bgp_default_peer_state
filter invalid_filter
{
  "rtbrick-config:filter": [
    {
      "match-attribute-name": "invalid_filter",
      "match-attribute-value": "Active",
```

```
      "match-type": "exact"
    }
  ]
}
```

**Error Type 2: Invalid Attribute**

In this configuration 'invalid_attribute' is not a valid attribute for the metric 'bgp_default_peer_state'. So, this is not a valid configuration.

```
set time-series metric bgp_default_peer_state attribute invalid_attribute
```

```
supervisor@rtbrick.net: cfg> show config time-series metric bgp_default_peer_state
attribute invalid_attribute
{
  "rtbrick-config:attribute": [
    {
      "attribute-name": "invalid_attribute"
    }
  ]
}
```

**Error Type 3: Invalid Attribute Type**

In this configuration, though 'af_configured' exists, it is not of a supported data type for metrics. So, this is not a valid configuration.

```
set time-series metric bgp_default_peer_state attribute af_configured
```

```
supervisor@rtbrick.net: cfg> show config time-series metric bgp_default_peer_state
attribute af_configured
{
  "rtbrick-config:attribute": [
    {
      "attribute-name": "af_configured"
    }
  ]
}
```

**Error Type 4: Invalid Label**

In this configuration the 'invalid_label' is not a valid label key. So this is not a valid configuration.

```
set time-series metric bgp_default_peer_state attribute state label invalid_label
```

```
{
  "rtbrick-config:label": [
    {
      "label-key": "invalid_label",
      "label-value": "invalid_label",
      "label-type": "dynamic"
    }
  ]
}
```

**Error Type 5: Invalid Label Type**

`error type5: invalid_label_attribute_type`

In this configuration, the label value ('af_configured') has a data type that is not allowed for label values. So this is not a valid configuration.

```
set time-series metric bgp_default_peer_state attribute state label invalid_label_type
set time-series metric bgp_default_peer_state attribute state label invalid_label_type
label-value af_configured
set time-series metric bgp_default_peer_state attribute state label invalid_label_type
label-type dynamic
```

```
supervisor@rtbrick.net: cfg> show config time-series metric bgp_default_peer_state
attribute state label invalid_label
_type
{
  "rtbrick-config:label": [
    {
      "label-key": "invalid_label_type",
      "label-value": "af_configured",
      "label-type": "dynamic"
    }
  ]
}
```

**Error Type 6: Invalid Filter Attribute Type**

`error type5: invalid_filter_attribute_type`

In this configuration, the filter value ('af_configured') has a data type that is not allowed for filter values. So this is not a valid configuration.

```
set time-series metric bgp_default_peer_state filter af_configured
set time-series metric bgp_default_peer_state filter af_configured match-attribute-
value ipv4-unicast
set time-series metric bgp_default_peer_state filter af_configured match-type exact
```

```
supervisor@rtbrick.net: cfg> show config time-series metric bgp_default_peer_state
filter af_configured
{
  "rtbrick-config:filter": [
    {
      "match-attribute-name": "af_configured",
      "match-attribute-value": "ipv4-unicast",
      "match-type": "exact"
    }
  ]
}
```

To see show command output for the error types for invalid configurations, see the section Show Command Output for Invalid Metric Configuration

#### Metric Profile Configuration

In a metric profile, you can club a collection of related metrics, which can be managed and exported together.

You can create a metric profile and associate a individual metric with the profile. It allows you to define a common setting such as scrape interval and append-timestamp for a group of metrics. This common settings will be applied automatically to all associated metrics. You can configure a maximum of 100 metric profiles.

> If you do not configure a Metric Profile, the system will use a default metric profile with a predefined scrape interval.

**Syntax:**

**set time-series metric-profile** <profile-name> **scrape-interval** <scrape-interval>

| Attribute | Description |
|---|---|
| `profile-name` | Specify the name of the profile. Note: Metric Profile names must not begin with the name 'rtbrick' as it is reserved for internal use. Similarly, RBFS has built-in default metric profiles, therefore, do not use the name 'default' as metric profile name. |
| `scrape-interval` | The frequency (period of time) at which the system fetches (scrapes) data from a target source. |
| `append-timestamp` | Add a time-stamp on every metric pulled. |

Example:

This configuration defines a metric profile called "up_metrics." The configuration sets the scrape interval to 10 seconds, meaning that metrics linked to this metric profile will be collected every 10 seconds.

```
set time-series metric-profile up_metrics
set time-series metric-profile up_metrics scrape-interval 10s
```

Example:

```
{
  "rtbrick-config:metric-profile": [
    {
      "profile-name": "up_metrics",
      "scrape-interval": "10s"
    }
  ]
}
```

#### Attaching a Metric with a Metric Profile

You can associate an existing metric with an existing metric profile.

**Syntax:**

**set time-series metric** <metric-name> **profile-name** <profile-name>

| Attribute | Description |
|---|---|
| `<metric-name>` | Specify the metric name. |
| `<profile-name>` | Specify the metric profile name. |

API To access the RESTCONF API that corresponds to this CLI, click here.

The following command is used to associate a metric named 'process_status' with the metric profile named 'up_metrics'.

```
set time-series metric process_status profile-name up_metrics
```

Example:

```
supervisor@rtbrick.net: cfg> show config time-series metric process_status profile-
name
{
  "rtbrick-config:profile-name": "up_metrics"
}
```

#### Alert Configuration

You can configure alerts which are notifications about the events based on the captured metrics.

Alerts cannot be assigned to an alert profile. It is the alert profiles that are used to forward alerts. For each alert, the system will utilize the first matching profile from the list of available alert profiles. The priority attribute of each profile determines the order in which alerts are processed.

**Syntax**

**set time-series alert-profile** <profile-name> <attribute> <value>

| Attribute | Description |
|---|---|
| `annotations` | Key-value pairs that add metadata to the alert. |
| `description` | Alert description. |
| `expr` | PromQL Alert evaluation. It defines condition for triggering this alert. |
| `for` | Specifies the duration the condition must be true before triggering the alert. |
| `group` | Logical grouping of alerts. |
| `interval` | Defines how often this alert expression (expr) is evaluated (for example, every 30s, 1m, and so on). |
| `labels` | Key-value pairs used to categorize alerts, mainly for routing in Alertmanager. |
| `level` | Alert's severity level. |
| `summary` | Explicit annotation label summary. |

To access the RESTCONF API that corresponds to this CLI, click here.

Example:

This configuration defines a time-series alert named 'chassis_fan_failure', which monitors the speed of chassis fans using time-series metrics. The alert continuously checks fan speeds and triggers if the fan speed remains below 1500 RPM for 5 consecutive minutes, with checks occurring every minute. It is organized under the group labeled "fans," which categorizes similar alerts.

The alert is assigned a severity level of 2 and the label "repeat_interval" is set to 24 hours, meaning the alert will not be resent more than once within that timeframe. Additional annotations include metadata such as the threshold (1500 RPM), the unit, the actual fan speed at the time of the alert, and the direction (below).

The alert only triggers if the condition holds true for a continuous 5-minute period. Specifically, it checks whether the maximum fan speed (in RPM) over the last 60 seconds is less than 1500. This alert serves to monitor for any fan failures, and if the speed is below the threshold for 5 minutes, it will trigger an alert.

```
set time-series alert chassis_fan_failure
set time-series alert chassis_fan_failure group fans
set time-series alert chassis_fan_failure for 5m
set time-series alert chassis_fan_failure interval 1m
set time-series alert chassis_fan_failure expr
"max_over_time(chassis_fan_speed_rpm[60s]) < 1500"
set time-series alert chassis_fan_failure level 2
set time-series alert chassis_fan_failure summary "Fan speed {{$value}}rpm of fan
{{$labels.fan}} is below 1500rpm."
set time-series alert chassis_fan_failure description "Fan speed is below 1500rpm"
set time-series alert chassis_fan_failure labels repeat_interval:24h
set time-series alert chassis_fan_failure annotations threshold:1500
set time-series alert chassis_fan_failure annotations unit:RPM
set time-series alert chassis_fan_failure annotations value:{{$value}}
set time-series alert chassis_fan_failure annotations "direction: below"
```

Example:

```
supervisor@rtbrick.net: cfg> show config time-series alert chassis_fan_failure
{
  "rtbrick-config:alert": [
    {
      "name": "chassis_fan_failure",
      "group": "fans",
      "for": "5m",
      "interval": "1m",
      "expr": "max_over_time(chassis_fan_speed_rpm[60s]) < 1500",
      "level": "2",
      "summary": "Fan speed {{$value}}rpm of fan {{$labels.fan}} is below 1500rpm.",
      "description": "Fan speed is below 1500rpm",
      "labels": [
        "repeat_interval:24h"
        ],
      "annotations": [
        "threshold:1500",
        "unit:RPM",
        "value:{{$value}}",
        "direction: below"
        ]
    }
  ]
}
```

#### Alert Profile Configuration

You can set up alert profiles for Prometheus time-series data. This configuration defines alerts, how alerts should be grouped, filtered, and repeated.

### Syntax

**set time-series alert-profile** <profile-name> <attribute> <value>

| Attribute | Description |
|---|---|
| `profile-name` | Specify the name of the alert profile. |
| `priority` | Specify the priority number. |
| `group-by` | Groups alerts by one or more labels. |
| `matchers` | Filters which alerts are processed by this profile. |
| `repeat-interval` | Defines how frequently notifications for unresolved alerts will be resent. |

API To access the RESTCONF API that corresponds to this CLI, click here.

The following configuration defines a time-series alert profile named '4h_receiver'. This profile specifies how alerts that meet certain criteria should be managed.

The priority level is set to '1', indicating that this alert profile has high priority, as lower numbers indicate higher preference. The 'repeat-interval' is configured for '4h', meaning that alerts associated with this profile will be resent every 4 hours if they remain active. In essence, alerts under this profile are sent once every 4 hours.

The 'matchers' parameter is defined as 'repeat_interval=4h', which filters the alerts to which this profile applies. Only alerts with the label 'repeat_interval=4h' will be managed by this profile.

```
set time-series alert-profile 4h_receiver
set time-series alert-profile 4h_receiver priority 1
set time-series alert-profile 4h_receiver repeat-interval 4h
set time-series alert-profile 4h_receiver matchers repeat_interval=4h
```

```
{
  "rtbrick-config:alert-profile": [
    {
      "profile-name": "4h_receiver",
      "priority": 1,
      "repeat-interval": "4h",
      "matchers": [
        "repeat_interval=4h"
        ]
    }
  ]
}
```

#### Enabling Time-series Query History

All BDS queries related to time-series exposure will be logged in a BDS table for review during issue

debugging. By default, time series database history is disabled.

**Syntax**

**set time-series history-status** [enable | disable]

| Attribute | Description |
|---|---|
| `[enable\|disable]` | Enable time series database history by specifying 'enable'. Time-series database history is disabled, by default. |

Example:

The following configuration enables timeseries history status by specifying 'enable'.

```
set time-series history-status enable
```

```
supervisor@rtbrick.net: cfg> show config time-series history-status
{
  "rtbrick-config:history-status": "enable"
}
```

#### Prometheus Self-Monitoring Configuration

You can enable a self-monitoring feature that allows Prometheus to monitor its own performance and health. This self-monitoring functionality collects metrics regarding the system's operation, such as query performance and resource usage. The scrape interval determines how often these metrics are collected.

**Syntax:**

**set time-series self-monitoring enable** <false|true> **set time-series self-monitoring scrape-interval** <scrape-interval[1-128]>

| Attribute | Description |
|---|---|
| `enable <false \| true>` | Specify true to enable the self monitoring functionality for Prometheus. |
| `<scrape-interval[1-128]>` | The frequency (period of time) at which the system fetches (scrapes) data from a target source. |

API To access the RESTCONF API that corresponds to this CLI, click here.

Example:

The following CLI commands enable the self-monitoring capability for the time-series database. The

self monitoring is enabled by specifying 'true'. The scrape-interval is defined as 10s. It means the metrics are collected every 10 seconds.

```
set time-series self-monitoring enable true
set time-series self-monitoring scrape-interval 10s
```

Example:

```
{
  "rtbrick-config:self-monitoring": {
    "enable": "true",
    "scrape-interval": "10s"
  }
}
```

#### Enabling Append-Timestamp in Prometheus

RBFS supports 'append-timestamp' functionality in Prometheus. When the append-timestamp flag is set, the first metric generated after an object is updated will include the actual timestamp of the update. This accurately reflects when the change took place. For all subsequent metrics for the same object (since it has not been been updated again) the current scrape time will be used.

You can enable `append-timestamp` on both metric or metric profile level. If you configure `append-timestamp` for both, metric-level gets priority.

**Syntax:**

**set time-series metric** <metric-name> **append-timestamp** <false|true> **set time-series metric-profile** <profile-name> **append-timestamp** <false|true>

| Attribute | Description |
|---|---|
| `<metric-name>` | Name of the metric. |
| `<profile-name>` | Name of the metric profile. |
| `append-timestamp <false / true>` | Specify true to enable time-stamp functionality. |

Example command:

The following command is used to add a timestamp to each metric associated with the metric profile called 'chassis'.

```
set time-series metric-profile chassis append-timestamp true
```

Example:

```
{
  "rtbrick-config:metric-profile": [
    {
      "profile-name": "chassis",
      "append-timestamp": "true"
    }
  ]
}
```

### 10.1.3. Observability Operational Commands

#### Show Commands

High-level time-series show command and options:

```
supervisor@rtbrick.net: cfg> show time-series
 alert
 metric
 statistics
```

##### Alert Details

Display information about the time-series alerts.

**Syntax:**

**show time-series alert** <alert-name>

| Option | Description |
|---|---|
| - | Without any option, the command lists all firing alerts. |
| <alert-name> | All alerts for the given alert rule. |
| pipe (\|) | Allows you to specify options: grep and count. |
| grep <grep_str> | Specify the grep string. |

Example: Information about time-series alert

```
supervisor@rtbrick.net: cfg> show time-series alert
State       Since                   Level        Summary
firing      26-JUN-2025 09:40:24    Critical     Process confd has crashed.
```

Example:

```
supervisor@rtbrick.net: cfg> show time-series alert | count
```

```
3
```

#### Metric Details

Display information about timeseries metrics.

**Syntax:**

**show time-series metric** <metric-name>

| Option | Description |
|---|---|
| `-` | Without any option, displays information about all metrics. |
| `rtbrick_invalid_metric_config` | Displays the list of misconfigurations in time-series. |

Example: Information for a specified metric.

```
supervisor@rtbrick.net: cfg> show time-series metric process_status
Metric: process_status
  Value: 1 sampled at 23-JUN-2025 10:20:08
  Labels:
    administrative_state          : UP
    bd                            : bgp.appd.1
    bd_name                       : etcd
    element_name                  : L1-STD-2-2008
    instance                      : localhost:1002
    job                           : bds_up_metrics
    pod_name                      : rtbrick-pod

  Value: 1 sampled at 23-JUN-2025 10:20:08
  Labels:
    administrative_state          : UP
    bd                            : bgp.iod.1
    bd_name                       : etcd
    element_name                  : L1-STD-2-2008
    instance                      : localhost:1002
    job                           : bds_up_metrics
    pod_name                      : rtbrick-pod

  Value: 1 sampled at 23-JUN-2025 10:20:08
  Labels:
    administrative_state          : UP
    bd                            : confd
    bd_name                       : etcd
    element_name                  : L1-STD-2-2008
    instance                      : localhost:1002
    job                           : bds_up_metrics
    pod_name                      : rtbrick-pod
```

```
  Value: 1 sampled at 23-JUN-2025 10:20:08
  Labels:
    administrative_state          : UP
    bd                            : fibd
    bd_name                       : etcd
    element_name                  : L1-STD-2-2008
```

**Invalid Metric Configuration**

Example for invalid metric configurations

NOTE:

In the output (as shown below), if the displayed 'value' continues to increase (incremental) with each subsequent output, it indicates that the issue is still unresolved. If the 'value' remains static (no longer increasing) in the subsequent outputs, it indicates that the issue has been resolved.

```
supervisor@rtbrick.net: cfg> show time-series metric rtbrick_invalid_metric_config
Metric: rtbrick_invalid_metric_config

  Value: 3 sampled at 30-JUL-2025 06:45:10
  Labels:
    attribute_name                : af_configured
    bd_name                       : bgp.iod.1
    element_name                  : L1-STD-1-1047
    error_type                    : invalid_attribute_type
    instance                      : localhost:3102
    job                           : bds_rtbrick_error
    metric_name                   : bgp_default_peer_state
    pod_name                      : rtbrick-pod
    profile_name                  : bgp_metrics

  Value: 21 sampled at 30-JUL-2025 06:45:10
  Labels:
    attribute_name                : af_configured
    bd_name                       : bgp.iod.1
    element_name                  : L1-STD-1-1047
    error_type                    : invalid_label_attribute_type
    instance                      : localhost:3102
    job                           : bds_rtbrick_error
    metric_name                   : bgp_default_peer_state
    pod_name                      : rtbrick-pod
    profile_name                  : bgp_metrics

  Value: 3 sampled at 30-JUL-2025 06:45:10
  Labels:
    attribute_name                : af_configured
    bd_name                       : bgp.iod.1
    element_name                  : L1-STD-1-1047
```

```
    error_type                    : invalid_filter_attribute_type
    instance                      : localhost:3102
    job                           : bds_rtbrick_error
    metric_name                   : bgp_default_peer_state
    pod_name                      : rtbrick-pod
    profile_name                  : bgp_metrics

  Value: 3 sampled at 30-JUL-2025 06:45:10
  Labels:
    attribute_name                : invalid_label
    bd_name                       : bgp.iod.1
    element_name                  : L1-STD-1-1047
    error_type                    : invalid_label_attribute
    instance                      : localhost:3102
    job                           : bds_rtbrick_error
    metric_name                   : bgp_default_peer_state
    pod_name                      : rtbrick-pod
    profile_name                  : bgp_metrics

  Value: 3 sampled at 30-JUL-2025 06:45:10
  Labels:
    attribute_name                : invalid_filter
    bd_name                       : bgp.iod.1
    element_name                  : L1-STD-1-1047
    error_type                    : invalid_filter_attribute
    instance                      : localhost:3102
    job                           : bds_rtbrick_error
    metric_name                   : bgp_default_peer_state
    pod_name                      : rtbrick-pod
    profile_name                  : bgp_metrics

  Value: 3 sampled at 30-JUL-2025 06:45:10
  Labels:
    attribute_name                : invalid_attribute
    bd_name                       : bgp.iod.1
    element_name                  : L1-STD-1-1047
    error_type                    : invalid_attribute
    instance                      : localhost:3102
    job                           : bds_rtbrick_error
    metric_name                   : bgp_default_peer_state
    pod_name                      : rtbrick-pod
    profile_name                  : bgp_metrics
```

**Time-series Statistics**

The `show time-series statistics` that provides visibility into the time-series processing on RBFS. It helps to identify any existing configuration errors and displays the number of active jobs currently, running to collect time-series data. The number of active jobs depends on the applied time-series metric configuration.

**Syntax:**

```
show time-series statistics
```

| Option | Description |
|---|---|
| `-` | Without any option, the command displays all statistics. |
| `pipe (\|)` | Allows you to specify options: grep and count. |
| `grep <grep_str>` | Specify the grep string. |

Example: Time-series statistics

```
supervisor@rtbrick.net: cfg> show time-series statistics
Scrape Targets
Metric Profile                        Daemons
bgp_metrics                           1
rtbrick_error                         1
up_metrics                            1

Error Counters
Name                                                                  Count
Increase (in last 5 mins)
prometheus_target_scrapes_sample_duplicate_timestamp_total         0           0
prometheus_target_scrapes_exemplar_out_of_order_total              0           0
prometheus_target_scrapes_sample_out_of_bounds_total               0           0
prometheus_target_scrapes_sample_out_of_order_total                0           0
prometheus_target_scrapes_exceeded_sample_limit_total              0           0
supervisor@L1-STD-2-2008>bm08-tst.hel.rtbrick.net: cfg>
```

# 10.2. Resmon

## 10.2.1. Resource Monitoring Overview

Monitoring the system resources is very crucial to analyze the health of devices. RBFS has dedicated daemons `peripherald` and `resmond` to discover and monitor the device resources.

RBFS Resource Monitoring functionality provides support for monitoring the following components:

- CPU
- Memory
- Processes
- Disks
- Sensor

- Transceivers

#### CPU

RBFS Resource Monitoring framework collects CPU hardware information from the `global.chassis_0.resource.cpu` table. In addition, it calculates CPU usage dynamically and stores this information in the `global.chassis_0.resource.cpu_usage` table.

#### Memory

RBFS Resource Monitoring feature collects RAM hardware information from the `global.chassis_0.resource.mem` table. It also gathers memory usage information in the `global.chassis_0.resource.mem_usage` table.

#### Processes

RBFS Resource Monitoring feature collects process usage information of Brick Daemon(BD) that runs in the RBFS and stores the information in the `global.chassis_0.resource.proc_usage` table. It dynamically gathers the process information and calculates the CPU and the memory usage of the individual Brick Daemons.

#### Disks

RBFS Resource Monitoring feature collects the disk information from the `global.chassis_0.resource.disk` table. It also collects disk usage information in the `global.chassis_0.resource.disk_usage` table.

#### Sensor

RBFS Resource Monitoring feature the reading data of the hardware sensor such as temperature, fan, power-supply, and system LED. The data collected from the sensor are stored in the following tables for the different components:

- `global.chassis_0.resource.sensor.chassis`
- `global.chassis_0.resource.sensor.psu`
- `global.chassis_0.resource.sensor.fan`
- `global.chassis_0.resource.sensor.thermal`
- `global.chassis_0.resource.sensor.led`

> The RBFS implementation supports pluggable transceiver modules on white box switches only.

#### System Clock

RBFS Resource Monitoring feature provides support for monitoring the system clock so that the system clock is always in sync with the NTP server clock. This ensures that the deviation from the NTP server clock always remains within acceptable limits. It collects the system clock information from the `global.os.timex` table.

> The RBFS implementation supports pluggable transceiver modules on white box switches only.

**Transceivers**

RBFS Resource Monitoring feature monitors transceivers plugged onto the chassis. It reads transceivers EEPROM (Electrically Erasable Programmable Read-Only Memory) data and translates the data to respective fields in the BDS tables.

RBFS provides the following functionalities for monitoring transceivers:

- Mechanism to discover and monitor transceiver modules. Supported transceiver modules include SFP, SFP+, QSFP, QSFP+, and QSFP28 (DAC is not supported).
- Provides CLIs to write to transceivers
- Provides show commands to see transceiver inventory and status of each module
- Logs the status of the transceivers

> The RBFS implementation supports monitoring of pluggable transceiver modules on white box switches only.

RBFS Resource Monitoring feature performs the following tasks:

- Transceiver: Identifying the following brief information of a discovered transceiver module and stores in table `global.transceiver`.
  - ifp_name
  - module_type
  - vendor_name
  - vendor_serial
  - vendor_part_number
- Read the following transceiver data from a module and stores in the table: `global.transceiver.lane`.
  - RX/TX alarming (loss of light and loss of signal)
- RX/TX power status
  - Voltage and BIAS status
  - Temperature
- Write the transceiver data to an transceiver module
  - Enabling high power class on QSFP28
  - Shutdown lasers (QSFP28, SFP+ and SFP)

**Logging**

RBFS Resource Monitoring feature can log the following transceiver module events:

- Temperature high alarm
- Temperature high warning
- Temperature low alarm
- Temperature low warning
- Voltage high alarm
- Voltage high warning
- Voltage low alarm
- Voltage low warning
- Lane power high alarm
- Lane power high warning
- Lane power low alarm
- Lane power low warning
- Lane bias high alarm
- Lane bias high warning
- Lane bias low alarm
- Lane bias low warning

**Q2C Resource Monitoring**

Q2C platform resource-specific usage metrics are stored in the BDS table: `local.bcm.resource.monitor`. Resource usage information enables you to understand the scale of services that the device performs and how it optimally leverages the resource usage.

The following table provides the list of supported resource types for monitoring in RBFS for Q2C platforms.

| Resource Type | Description |
|---|---|
| EEDB_ARP | EEDB is an Egress Encapsulation Data Base. This resource is consumed when ARP (Address Resolution Protocol) entries. |
| EEDB_L2TP | EEDB is an Egress Encapsulation Data Base. This resource is consumed when L2TP subscribers are created in hardware. |
| EEDB_MPLS_TUNNEL | EEDB is an Egress Encapsulation Data Base. This resource is consumed when MPLS tunnels are created in the chip. |
| EEDB_PPPOE | EEDB_PPPOE is used for PPPoE encapsulation. This resource is consumed when PPPoE subscribers are created in hardware. |
| EEDB_PWE | This resource is consumed when L2X or cross-connection sessions are created in hardware at egress. |
| IN_LIF_FORMAT_PWE | This resource is consumed when a pseudowire is created in the hardware at ingress. |

| Resource Type | Description |
|---|---|
| IN_AC_C_C_VLAN_DB | This resource is consumed when an ingress logical interface for double-tagged VLAN is created. |
| IN_AC_C_VLAN_DB | This resource is consumed when an ingress logical interface for single tag VLAN is created. |
| IN_AC_UNTAGGED_DB | This resource is consumed when an ingress logical interface for untagged IFLs is created. |
| IPV4_MULTICAST_PRIVATE_LPM_FORWARD | LPM stands for Longest Prefix Match. This resource is consumed for multicast (source, group) entries. |
| IPV4_UNICAST_PRIVATE_LPM_FORWARD | This resource is consumed for both default and non-default VRF instance IPv4 prefixes. |
| IPV6_UNICAST_PRIVATE_LPM_FORWARD | This resource is consumed for both default and non-default VRF instance IPv6 prefixes. |
| L3_RIF | This resource is consumed for the L3 interfaces. |
| L2TPV2_DATA_MESSAGE_TT | This resource is consumed when the L2TP subscribers are created in hardware. |
| MPLS_FWD | This resource is consumed for MPLS entries for which forwarding actions are involved. |
| MPLS_TERMINATION_SINGLE_LABEL_DB | This resource is consumed for MPLS entries for which label termination is required. |
| MULTICAST_MCDB | This resource is consumed for multicast groups created in hardware. |
| PPPOE_O_ETH_TUNNEL_FULL_SA | This resource is consumed for PPPoE subscribers in hardware. |

Example:

```
supervisor@rtbrick.net: cfg> show hardware resource monitor
  Resource Type                                          Consumed Estimated
  ECMP_GROUP                                                    0     40960
  EEDB_ARP                                                     27    114688
  EEDB_L2TP                                                     0    114688
  EEDB_MPLS_TUNNEL                                             11    114688
  EEDB_PPPOE                                                    0    114688
  EEDB_PWE                                                      0    114688
  IN_AC_C_C_VLAN_DB                                             0     60292
  IN_AC_C_VLAN_DB                                              15     60292
  IN_AC_S_C_VLAN_DB                                             0     60292
  IN_AC_S_VLAN_DB                                               0     60292
  IN_AC_UNTAGGED_DB                                             1     60292
  IN_LIF_FORMAT_PWE                                             0    131072
  IPV4_MULTICAST_PRIVATE_LPM_FORWARD                            0    284160
  IPV4_UNICAST_PRIVATE_LPM_FORWARD                              9    568320
  IPV4_UNICAST_PRIVATE_LPM_FORWARD_2                            0    284160
```

```
    IPV6_UNICAST_PRIVATE_LPM_FORWARD                          9    284160
    IPV6_UNICAST_PRIVATE_LPM_FORWARD_2                        0    142080
    KBP_IPV4_UNICAST_PRIVATE_LPM_FORWARD                      0         0
    KBP_IPV6_UNICAST_PRIVATE_LPM_FORWARD                      0         0
    KBP_NAT_EXTERNAL_TO_INTERNAL                              0         0
    KBP_NAT_EXTERNAL_TO_INTERNAL_STATS                        0         0
    KBP_NAT_INTERNAL_TO_EXTERNAL                              0         0
    KBP_NAT_INTERNAL_TO_EXTERNAL_STATS                        0         0
    L2TPV2_DATA_MESSAGE_TT                                    0     61275
    L3V4_EXTERNAL                                             8      1024
    L3V6_EXTERNAL                                             7      1024
    L3_RIF                                                   22     40959
    MFV4_EXTERNAL                                             0      2048
    MFV6_EXTERNAL                                             0      1024
    MPLS_FWD                                                  6   1017111
    MPLS_TERMINATION_SINGLE_LABEL_DB                          8     61275
    MULTICAST_MCDB                                            2    262142
    POLICER_EXTERNAL                                          0      4096
    PPPOE_O_ETH_TUNNEL_FULL_SA                                0     27361
    QOS_CL_ELEMENTS                                           0     30000
    QOS_FQ_ELEMENTS                                           0     30000
    QOS_HR_ELEMENTS                                         168     30000
    QOS_MC_QUEUE_BUNDLES                                      1     32000
    QOS_SCH_SHARED_SHAPER_DUAL                                0     30000
    QOS_SCH_SHARED_SHAPER_OCTA                                0     30000
    QOS_SCH_SHARED_SHAPER_QUAD                                0     30000
    QOS_UC_QUEUE_BUNDLES                                    136     32000
    QOS_VOQ_CONNECTOR_BUNDLES                                52     32000
```

#### QAX Resource Monitoring

QAX platform resource-specific usage metrics are stored in the BDS table: `local.bcm.resource.monitor`. Resource usage information enables you to understand the scale of services that the device performs and how it optimally leverages the resource usage.

The following table provides the list of supported resource types for monitoring in QAX platforms.

| Resource Type | Description |
|---|---|
| EEDB_ARP | EEDB is an Egress Encapsulation Data Base. This resource is consumed when ARP (Address Resolution Protocol) entries. |
| ECMP ID | It represents an index in the table that maintains the paths for a particular ECMP route. |
| Egress Failover ID | It identifies failover paths for traffic leaving the device. |
| FEC Failover ID | This is used to manage failover paths for FECs. |
| FECs for Global Use | It represents the number of FECs that are available for general use across the device. |

| Resource Type | Description |
| --- | --- |
| Field Direct Extraction Entry ID | It represents the entries in the FDE table, which are used for direct extraction of fields. |
| Field Entry ID | It identifies a specific entry used for field-based operations. |
| Ingress Failover ID | It identifies backup paths for traffic entering the device. |
| Local Common InLif | It represents the number of common InLifs (Ingress Logical Interface) allocated for incoming traffic. |
| Local OutLif | It represents the number of logical interfaces allocated for outgoing traffic. |
| Local Wide InLif | Wide InLifs are used for additional metadata or special handling is required for incoming traffic. It represents the number of wide InLifs allocated for such purposes. |
| Number of Meters in Processor A | Meters are used for rate limiting or policing traffic, ensuring that traffic adheres to predefined bandwidth limits. This represents the number of meters available and used in processor A. |
| Profiles for PON Use | Profiles related to Passive Optical Networks are used in the management and configuration of PON interfaces. The number of profiles allocated for PON operations. |
| QOS EGRESS DSCP/EXP MARKING PROFILE IDs | These IDs represent the QoS marking profiles used to mark outgoing traffic based on DSCP or EXP bits, ensuring the prioritization of outgoing traffic. |
| QOS EGRESS L2 I TAG PROFILE IDs | These IDs are used for profiles that manage Layer 2 (L2) tagging in egress traffic. The number of available profiles for L2 I Tagging. |
| QOS EGRESS MPLS PHP QOS IDs | The IDs identify QoS profiles used for MPLS PHP (Penultimate Hop Popping) operations. |
| QOS EGRESS REMARK QOS IDs | This is used to re-mark packets with a new QoS value as they exit the device. IDs represent the QoS profiles that apply new markings to egress traffic. |
| QOS INGRESS COS OPCODE IDs | COS (Class of Service) opcodes are used to classify incoming traffic based on criteria such as type of service. The IDs represent different COS opcode profiles for classifying ingress traffic. |
| QOS INGRESS LABEL MAP ID | Maps are used to associate incoming packets with certain labels for QoS processing. The number of available Label Maps for ingress traffic. |
| QOS INGRESS LIF/COS IDs | This ID is used for associating incoming traffic with specific LIF (Logical Interface) and COS profiles for handling. The number of IDs available for mapping ingress traffic to LIF/COS profiles. |
| QOS INGRESS PCP PROFILE IDs | Priority Code Point (PCP) profiles are used to prioritize packets at the ingress based on their PCP value. The IDs represent available PCP profiles for prioritizing ingress traffic. |

| Resource Type | Description |
|---|---|
| SW Handles of Policer | It represents the number of software handles available for configuring policers. |
| Trill Virtual Nickname | It represents the number of virtual nicknames allocated for TRILL protocol operations for creating loop-free multi-path Ethernet network. |
| VLAN Translation Egress Usage | The number of entries used for VLAN translation operations on out-going traffic. |
| VLAN Translation Ingress Usage | The number of entries used for VLAN translation on incoming traffic. |
| VSIs for MSTP | The number of VSIs (Virtual Switch Instances) allocated for MSTP operations. |
| VSIs for TB VLANS | The number of VSIs allocated for handling TB VLANs. |

Example:

The following is an example output for the `show hardware resource monitor` command from the UfiSpace S9500-22XST Multiservice Edge Platform.

```
supervisor@rtbrick.net: cfg> show hardware resource monitor
  Resource Type                                        Consumed Estimated
  Cross Connect                                            4000     20000
  Double tagged LIF                                           0     65000
  ECMP id                                                     0      4095
  Egress Failover id                                          0     32767
  FEC Failover id                                             0     32765
  FECs for global use                                      2009     61440
  Field entry id                                              2     65538
  Field extraction entry id                                  15        64
  IPv4 UC KAPS                                                0     50000
  IPv4 UC LEM                                                 1     50000
  IPv6 UC Route                                               3     20000
  Ingress Failover id                                         0     32767
  Local Common InLif                                       5011     65536
  Local OutLif                                            10033    163840
  Local Wide InLif                                         3341     43690
  MPLS LSR LEM                                             2000     20000
  MPLS SEMA                                                   0     20000
  MPLS SEMB                                                   0     20000
  Number of meters in processor A                             0     32768
  Number of meters in processor B                             0     32768
  Profiles for PON use                                        0        16
  QOS EGRESS DSCP/EXP MARKING PROFILE IDs                     0        16
  QOS EGRESS L2 I TAG PROFILE IDs                             0         1
  QOS EGRESS MPLS PHP QOS IDs                                 0         3
  QOS EGRESS REMARK QOS IDs                                   3        15
  QOS INGRESS COS OPCODE IDs                                  0         7
```

```
QOS INGRESS LABEL MAP ID                                  0         1
QOS INGRESS LIF/COS IDs                                   1        63
QOS INGRESS PCP PROFILE IDs                               0        15
SW handles of policer                                     0         7
Single tagged LIF                                      2004     65000
Trill virtual nickname                                    0         2
Untag LIF                                                 0     65000
VLAN translation egress usage                            14       512
VLAN translation ingress usage                           11        64
VSIs for MSTP                                            16     28672
VSIs for TB VLANS                                         1      4096
```

#### Supported Platforms

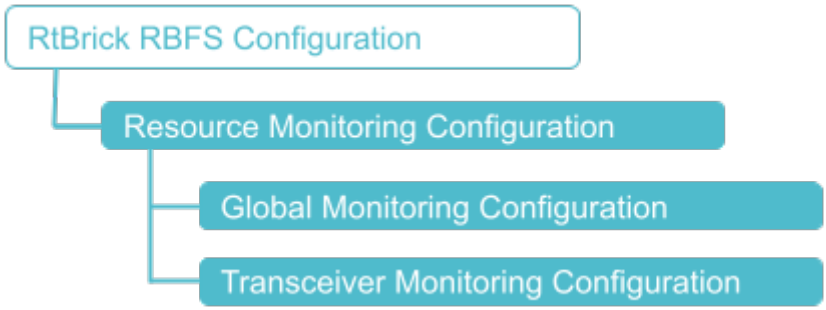
Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

### 10.2.2. Resource Monitoring Configuration

#### Configuration Hierarchy

The diagram illustrates the resource monitoring configuration hierarchy.

*Optic Configuration Hierarchy*




#### Configuration Syntax and Commands

The following sections describe the configuration syntax and commands.

##### Global Monitoring Configuration

This command sets the poll-interval for the resource monitoring functionality to discover transceivers.

**Syntax:**

**set resmon monitoring** <poll-interval>

| Attribute | Description |
| --- | --- |
| `poll_interval <poll-interval>` | Specifies the interval (in seconds) at which transceiver should be polled. The interval can range from 3 to 10000 seconds. The default interval is 10 seconds. |

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Transceiver Configuration

You can use this command to disable or enable (By, default enabled) tx laser or high-power class of an transceiver module on a specific interface.

**Syntax:**

**set resmon optics** <interface> ...

| Attribute | Description |
| --- | --- |
| `interface <interface-name>` | Name of the interface |
| `high-power-class [disable / enable]` | Enable or disable high power class for transceiver module. Enabled, by default. |
| `tx [disable / enable]` | Enable or disable lasers for transceiver module. Enabled, by default. |

API To access the RESTCONF API that corresponds to this CLI, click here.

#### QoS Resource Monitoring

RBFS provides resource monitoring support for Quality of Service (QoS) components. It provides visibility into current usage and maximum capacity for various QoS components.

Resource monitoring is enabled for the following QoS components:

- QOS_CL_ELEMENTS
- QOS_FQ_ELEMENTS
- QOS_HR_ELEMENTS
- QOS_MC_QUEUE_BUNDLES
- QOS_SCH_SHARED_SHAPER_DUAL
- QOS_SCH_SHARED_SHAPER_OCTA
- QOS_SCH_SHARED_SHAPER_QUAD
- QOS_UC_QUEUE_BUNDLES
- QOS_VOQ_CONNECTOR_BUNDLES

Example:

```
supervisor@rtbrick>LEAF01: cfg> show hardware resource monitor
<...>
  QOS_CL_ELEMENTS                                          0     30000
  QOS_FQ_ELEMENTS                                          0     30000
  QOS_HR_ELEMENTS                                        170     30000
  QOS_MC_QUEUE_BUNDLES                                     1     32000
  QOS_SCH_SHARED_SHAPER_DUAL                           12004     30000
  QOS_SCH_SHARED_SHAPER_OCTA                               0     30000
  QOS_SCH_SHARED_SHAPER_QUAD                               0     30000
  QOS_UC_QUEUE_BUNDLES                                 12106     32000
  QOS_VOQ_CONNECTOR_BUNDLES                            12021     32000
```

#### API Endpoint to Perform Power-Cycle Operation

You can use the API endpoint to perform the power-cycle operation. The following is the CTRLD API endpoint to perform the power-cycle operation.

POST /api/v1/ctrld/system/_powercycle

As shown in the example below, you can use CURL to invoke the power cycle function.

```
curl -i -H "Content-Type: application/json" -X POST http://<ip of
router>:19091/api/v1/ctrld/system/_powercycle
```

In the above URL, 19091 is a port number. This is the only supported port for this.

### 10.2.3. Resource Monitoring Operational Commands

#### Resource Monitoring Show Commands

The Resource Monitoring show commands provide detailed information about the resources and their usage.

#### CPU Information

This command displays the CPU detail and CPU information.

**Syntax:**

**show cpu** <option>

| Option | Description |
|---|---|
| `summary` | Displays the CPU information. |
| `usage` | Displays the CPU usage information |

Example: CPU summary

```
supervisor@rtbrick: op> show memory summary
System Memory
  Maximum capacity                 : 256 GB
  Error correction type            : Single-bit ECC
  Number of memory slots available : 4
  Number of memory slots occupied  : 2
Bank              Size      Location          Type    Speed       Configured Speed
Vendor            Serial No   Part No
NODE 1            16 GB     CPU1_DIMM_A1      DDR4    2666 MT/s   2133 MT/s
<BAD INDEX>       <BAD INDEX> <BAD INDEX>
NODE 1            16 GB     CPU1_DIMM_B1      DDR4    2666 MT/s   2133 MT/s
<BAD INDEX>       <BAD INDEX> <BAD INDEX>
```

Example: CPU usage information.

```
supervisor@rtbrick: op> show cpu usage
Name         Total       User     System       Nice   I/O Wait       Idle        IRQ   Soft IRQ
cpu            31%        23%         8%         0%         0%        68%         0%         0%
cpu0           11%         4%         5%         0%         0%        88%         0%         2%
cpu1          100%        63%        36%         0%         0%         0%         0%         0%
cpu2           54%        51%         2%         0%         0%        45%         0%         0%
cpu3           57%        55%         2%         0%         0%        42%         0%         0%
cpu4            6%         1%         5%         0%         0%        93%         0%         0%
cpu5            4%         2%         2%         0%         0%        95%         0%         0%
cpu6            8%         3%         4%         0%         0%        91%         0%         0%
cpu7            6%         1%         5%         0%         0%        94%         0%         0%
supervisor@rtbrick: op>
```

### Memory Details

This command displays the memory details and usage information.

**Syntax:**

**show memory** <option>

| Option | Description |
|---|---|
| `summary` | Displays the system memory information. |
| `usage` | Displays the memory usage information. |

Example: System memory information

```
supervisor@rtbrick: op> show memory summary
System Memory
```

```
  Maximum capacity                : 256 GB
  Error correction type           : Single-bit ECC
  Number of memory slots available : 4
  Number of memory slots occupied  : 2
Bank              Size      Location          Type    Speed       Configured Speed
Vendor            Serial No   Part No
NODE 1            16 GB     CPU1_DIMM_A1      DDR4    2666 MT/s   2133 MT/s
<BAD INDEX>       <BAD INDEX> <BAD INDEX>
NODE 1            16 GB     CPU1_DIMM_B1      DDR4    2666 MT/s   2133 MT/s
<BAD INDEX>       <BAD INDEX> <BAD INDEX>
```

Example: System memory usage information.

```
supervisor@rtbrick: op> show memory usage
Name     Total            Used             Free         Shared       Buffers
Cached
RAM      31.01 GiB        2.23 GiB         27.85 GiB    18.37 MiB    0 bytes
953.65 MiB
SWAP     0 bytes          0 bytes          0 bytes      n/a          n/a          n/a
```

**Process Details**

This command displays the process usage information.

**Syntax:**

**show process usage** <option>

| Option | Description |
|---|---|
| process-id <pid> | Displays the process usage for the specified process identifier. |
| process-name <process-name> | Displays the process usage for the specified process. |
| summary | Displays the process usage summary information. |

Example: Process usage summary information.

```
supervisor@rtbrick: op> show process usage summary
Name                    PID    Virtual Memory   Resident Memory     Shared Memory
CPU %     Mem %  CPU Core
bgp.appd.1             1027        305.68 MiB         97.42 MiB         99.35 MiB
0.20      0.31  0-15
bgp.iod.1              1029        530.68 MiB         97.25 MiB         99.35 MiB
0.20      0.31  0-15
confd                   210        314.10 MiB        145.29 MiB        103.86 MiB
1.83      0.46  0-15
etcd                    183         55.70 MiB         42.41 MiB         14.31 MiB
0.00      0.13  0-15
```

```
fibd                    345         13.73 GiB          1.55 GiB         93.39 MiB
81.05      4.98  0-15
ifmd                    328        206.04 MiB         47.89 MiB         14.58 MiB
0.41      0.15  0-15
isis.appd.1            1031        280.76 MiB         91.17 MiB         98.21 MiB
0.20      0.29  0-15
isis.iod.1             1035        413.43 MiB         91.99 MiB         98.21 MiB
0.20      0.29  0-15
lldpd                   307        272.14 MiB         93.01 MiB         97.79 MiB
1.83      0.29  0-15
mribd                   309        263.56 MiB         90.13 MiB         97.79 MiB
0.00      0.28  0-15
oamd                    188        312.15 MiB         44.35 MiB         14.38 MiB
0.00      0.14  0-15
ospf.appd.1            1038        265.06 MiB         92.14 MiB         98.26 MiB
0.00      0.29  0-15
ospf.iod.1             1040        506.62 MiB         91.76 MiB         98.26 MiB
0.20      0.29  0-15
policy.server          1044        247.48 MiB         89.30 MiB         97.75 MiB
0.00      0.28  0-15
poold                   193         54.13 MiB         40.38 MiB         14.32 MiB
0.00      0.13  0-15
resmond                 203        342.52 MiB         41.62 MiB         14.45 MiB
0.20      0.13  0-15
ribd                    311        354.05 MiB         99.52 MiB         97.89 MiB
0.00      0.31  0-15
staticd                 315         70.21 MiB         41.84 MiB         14.31 MiB
0.00      0.13  0-15
```

Example: Process usage for the specified process.

```
supervisor@rtbrick: op> show process usage process-name fibd
Process Name: fibd
  PID                          : 345
  REST port                    : 5522
  Debug port                   : 0
  Allowed CPU list             : 0-15
  CPU usage at user space      : 42249
  CPU usage at kernel space    : 5385
  CPU usage percentage         : 81.048746
  Memory usage percentage      : 4.982259
  Peak virtual memory usage    : 14395060
  Current virtual memory usage : 13.73 GiB
  Locked virtual memory        : 0 bytes
  RSS virtual memory           : 1.55 GiB
  Size of data segment         : 3.22 GiB
  Size of stack                : 132 KiB
  Size of text segment         : 4 KiB
  Shared library memory usage  : 93.39 MiB
  Swap space used              : 0 bytes
```

```
  Peak RSS                      : 1.55 GiB
  Page table entry              : 3.93 MiB
  Process state                 : ACTIVE
Module Status:
  bd     : Active
  bds    : Active
```

Example: Process usage for the specified process ID.

```
supervisor@rtbrick: op> show process usage process-id 1027
Process Name: bgp.appd.1
  PID                           : 1027
  REST port                     : 4102
  Debug port                    : 0
  Allowed CPU list              : 0-15
  CPU usage at user space       : 187
  CPU usage at kernel space     : 54
  CPU usage percentage          : 0.000000
  Memory usage percentage       : 0.306776
  Peak virtual memory usage     : 313012
  Current virtual memory usage : 305.68 MiB
  Locked virtual memory         : 0 bytes
  RSS virtual memory            : 97.42 MiB
  Size of data segment          : 38.07 MiB
  Size of stack                 : 132 KiB
  Size of text segment          : 4 KiB
  Shared library memory usage   : 99.35 MiB
  Swap space used               : 0 bytes
  Peak RSS                      : 97.42 MiB
  Page table entry              : 404 KiB
  Process state                 : ACTIVE
Module Status:
  bd     : Active
  bds    : Active
```

**Sensor Details**

This command displays the fan, power-supply, and temperature information.

**Syntax:**

**show sensor** <option>

| Option | Description |
|---|---|
| `fan` | Displays information about the sensor fan. |
| `power-supply` | Displays the sensor power supply information. |

| Option | Description |
|---|---|
| `temperature` | Displays the sensor temperature information. |

## Sensor Information

The `show sensor` command displays information about fan parameters, power-supply unit parameters, thermal sensors present on the system.

**Syntax:**

**show sensor [power-supply | fan | temperature]**

| Option | Description |
|---|---|
| `fan` | Displays information about fan parameters. |
| `power-supply` | Displays information about power-supply unit parameters. |
| `temperature` | Displays information about thermal sensors present on the system. |

Example: Summary information for sensor fan.

```
supervisor@rtbrick: op> show sensor fan
Name: CHASSIS FAN 0, Location: Fan Board
Status: -|PRESENT|-|-|F2B|-, BMC status: ok
  Attribute       Value  Low Warn  Upper Warn  Low Crit  Upper Crit  Low Shut  Upper
Shut
  Speed(RPM)     5700.0      2500         n/a      1500         n/a       500
n/a
Name: CHASSIS FAN 1, Location: Fan Board
Status: -|PRESENT|-|-|F2B|-, BMC status: ok
  Attribute       Value  Low Warn  Upper Warn  Low Crit  Upper Crit  Low Shut  Upper
Shut
  Speed(RPM)     5600.0      2500         n/a      1500         n/a       500
n/a
Name: CHASSIS FAN 2, Location: Fan Board
Status: -|PRESENT|-|-|F2B|-, BMC status: ok
  Attribute       Value  Low Warn  Upper Warn  Low Crit  Upper Crit  Low Shut  Upper
Shut
  Speed(RPM)     5700.0      2500         n/a      1500         n/a       500
n/a
Name: CHASSIS FAN 3, Location: Fan Board
Status: -|PRESENT|-|-|F2B|-, BMC status: ok
  Attribute       Value  Low Warn  Upper Warn  Low Crit  Upper Crit  Low Shut  Upper
Shut
  Speed(RPM)     5700.0      2500         n/a      1500         n/a       500
n/a
Name: PSU 0 FAN, Location: PSU 0
```

```
Status: -|PRESENT|-|-|F2B|-, BMC status: ok
  Attribute       Value  Low Warn  Upper Warn  Low Crit  Upper Crit  Low Shut  Upper
Shut
  Speed(RPM)    18000.0       n/a         n/a       n/a         n/a      1152
n/a
Name: PSU 1 FAN, Location: PSU 1
Status: -|PRESENT|FAILED|-|F2B|-, BMC status: Lower Non-Recoverable
  Attribute       Value  Low Warn  Upper Warn  Low Crit  Upper Crit  Low Shut  Upper
Shut
  Speed(RPM)        0.0       n/a         n/a       n/a         n/a      1152
n/a
```

Example: Sensor temperature

```
supervisor@rtbrick: op> show sensor temperature
Name                   Location    Temp(°C)  Low Warn  Upper Warn  Low Crit  Upper
Crit  Low Shut  Upper Shut  Status
CPU Package Thermal    CPU             35.0       n/a         n/a       n/a
85       n/a          95  -|PRESENT|-|-|-|-
PSU 0 THERMAL 1        PSU 0           32.0       n/a         n/a       n/a
65       n/a          70  -|PRESENT|-|-|-|-
PSU 1 THERMAL 1        PSU 1           31.0       n/a         n/a       n/a
65       n/a          70  -|PRESENT|-|-|-|-
TEMP_CPU_PECI          CPU Core        40.0       n/a          85       n/a
95       n/a         100  -|PRESENT|-|-|-|-
TEMP_ENV0              Main Board      28.0       n/a          60       n/a
65       n/a          70  -|PRESENT|-|-|-|-
TEMP_ENV1              Main Board      32.0       n/a          60       n/a
65       n/a          70  -|PRESENT|-|-|-|-
TEMP_ENV_CPU           Main Board      33.0       n/a          85       n/a
95       n/a         100  -|PRESENT|-|-|-|-
TEMP_ENV_MAC0          Main Board      32.0       n/a          65       n/a
70       n/a          80  -|PRESENT|-|-|-|-
TEMP_MAC0              Main Board      60.0       n/a         105       n/a
110       n/a         115  -|PRESENT|-|-|-|-
```

Example: Power-supply unit parameters details

```
supervisor@rtbrick: op> show sensor power-supply
Name: PSU 0, Manufacturer: ACBEL, Product name: FSH082, Part number: FSH082-HAAG,
Serial number: A13200S90420W501624
Status: -|PRESENT|-|-|-|-, BMC status: OK
  Attribute       Value  Low Warn  Upper Warn  Low Crit  Upper Crit  Low Shut  Upper
Shut
  IIn(Amps)        1.17       n/a         n/a       n/a         n/a      0.00
11.97
  IOut(Amps)       18.9       n/a         n/a       n/a         n/a      0.00
83.16
  VIn(Volts)      224.4       n/a         n/a       n/a         n/a     90.00
```

```
264.00
  VOut(Volts)      12.0       n/a         n/a       n/a         n/a     11.40
12.60
  PIn(Watts)      263.2       n/a         n/a       n/a         n/a      0.00
n/a
  POut(Watts)     229.6       n/a         n/a       n/a         n/a       n/a
1002.40
Name: PSU 1, Manufacturer: ACBEL, Product name: FSH082, Part number: FSH082-HAAG,
Serial number: A13200S90420W501636
Status: -|PRESENT|FAILED|-|-|-, BMC status: DC_POWER_FAILED
  Attribute       Value  Low Warn  Upper Warn  Low Crit  Upper Crit  Low Shut  Upper
Shut
  IIn(Amps)         0.0       n/a         n/a       n/a         n/a      0.00
11.97
  IOut(Amps)        0.0       n/a         n/a       n/a         n/a      0.00
83.16
  VIn(Volts)        0.0       n/a         n/a       n/a         n/a     90.00
264.00
  VOut(Volts)       0.0       n/a         n/a       n/a         n/a     11.40
12.60
  PIn(Watts)        0.0       n/a         n/a       n/a         n/a      0.00
n/a
  POut(Watts)       0.0       n/a         n/a       n/a         n/a       n/a
1002.40
```

**System LED Details**

Display information about system LED parameters.

**Syntax:**

```
show system-led
```

```
supervisor@rtbrick.net: op> show system-led
Name                      Colour  Blinking  Status
CHASSIS LED 1 (SYNC LED)  green   disabled  -|PRESENT|-|-|-
CHASSIS LED 2 (SYS LED)   green   disabled  -|PRESENT|-|ON|-
CHASSIS LED 3 (FAN LED)   green   disabled  -|PRESENT|-|ON|-
CHASSIS LED 4 (PSU0 LED)  green   disabled  -|PRESENT|-|ON|-
CHASSIS LED 5 (PSU1 LED)  yellow  enabled   -|PRESENT|-|ON|-
```

**Transceiver Details**

These show commands display information about transceivers.

**Syntax:**

**show transceiver** [<interface-name> | detail ]

| Option | Description |
|---|---|
| - | Without any option, the command displays the information for all transceivers. |
| <interface-name> | Displays detailed information for the transceiver. |
| detail | Displays the detailed view for a specified transceiver or all transceivers. |

Example (show transceiver) command output> It displays all transceiver information

```
supervisor@rtbrick.net: op> show transceiver
Interface      Module Type  Transceiver Name    Connector Type           PC         Vendor        Part Number
Serial Number     Material Number
ifp-0/0/1      QSFP28       100G-CWDM4          LC (Lucent Connector)    4(3.5W)    FS            QSFP28-IR4-100G
F2030883005       CMUIAMACAB10-3146-02
ifp-0/0/2      QSFP28       100GBASE-CR4        No separable connector   n/a        FS            Q28-PC02
G2006417081-2     n/a
ifp-0/0/3      QSFP28       100G-CWDM4          LC (Lucent Connector)    4(3.5W)    FS            QSFP28-IR4-100G
F2030882983       CMUIAMACAB10-3146-02
ifp-0/1/4      SFP          25GBASE-CR          Copper Pigtail           n/a        FS            S28-PC01
G2110077546-1     n/a
ifp-0/1/5      SFP          1GBASE-T            n/a                      n/a        FS            SFP-GB-GE-T
F1940056064       n/a
ifp-0/1/6      SFP          1GBASE-T            n/a                      n/a        FS            SFP-GB-GE-T
G1809118354       n/a
ifp-0/1/7      SFP          10GBASE-CR Passive  Copper Pigtail           n/a        FS            SFPP-PC015
S2108004622-2     n/a
ifp-0/1/8      SFP          10GBASE-CR Passive  Copper Pigtail           n/a        FS            SFPP-PC02
F2011245178-1     n/a
ifp-0/1/9      SFP          10GBASE-CR Passive  Copper Pigtail           n/a        FS            SFPP-PC02
F2011245179-1     n/a
ifp-0/1/10     SFP          10GBASE-CR Passive  Copper Pigtail           n/a        FS            SFPP-PC02
F2011245154-1     n/a
ifp-0/1/11     SFP          10GBASE-CR Passive  Copper Pigtail           n/a        FS            SFPP-PC01
F2032547869-2     n/a
ifp-0/1/12     SFP          10GBASE-CR Passive  Copper Pigtail           n/a        FS            SFPP-PC01
F2032547859-2     n/a
ifp-0/1/13     SFP          10GBASE-CR Passive  Copper Pigtail           n/a        FS            SFPP-PC01
F2032547887-1     n/a
ifp-0/1/14     SFP          10GBASE-LR          LC (Lucent Connector)    1(1.5W)    FS            SFP-10GLR-31
F1930111179       n/a
ifp-0/1/15     SFP          10GBASE-CR Passive  Copper Pigtail           n/a        FS            SFPP-PC01
F2032547865-2     n/a
ifp-0/1/16     SFP          25GBASE-CR          Copper Pigtail           n/a        FS            S28-PC015
S2108224439-1     n/a
ifp-0/1/17     SFP          25GBASE-CR          Copper Pigtail           n/a        FS            S28-PC015
S2108224426-1     n/a
<...>
```

Example output for a specified transceiver.

```
supervisor@rtbrick: op> show transceiver ifp-0/1/22
ifp-0/1/22
  Transceiver name: 10GBASE-CR Passive, Vendor name: FS, Module type: SFP,  Serial
```

```
number: C2401249812-2
  Part number: SFPP-PC025, Material number: n/a
  Link length support: 3m, Connector type: Copper Pigtail
  Manufactured date: Tue Jan 16 00:00:00 GMT +0000 2024
```

The following is a show command output displays an indicator for the specified transceiver 'ifp-0/1/10'.

```
supervisor@rtbrick.net: cfg> show transceiver ifp-0/1/10
ifp-0/1/10
  Transceiver name: n/a, Vendor name: n/a, Module type: Unknown(Transceiver-EEPROM
corruption/read-failure), Serial number: n/a
  Part number: n/a, Material number: n/a
  Power class: n/a
  Connector type: n/a, Link length support: n/a  Manufactured date: n/a
```

An error indicator in the output flags an error message for broken or faulty transceivers or when the transceiver EEPROM is corrupted or cannot be read. It allows for quick fault identification without the need for additional diagnostic steps.

Example for detailed view for the specified transceiver interface.

```
supervisor@rtbrick: op> show transceiver ifp-0/1/22 detail
Interface name: ifp-0/1/22
  Transceiver name: 10GBASE-CR Passive, Vendor name: FS, Module type: SFP
  Serial number: C2401249812-2, Part number: SFPP-PC025, Material number: n/a
  Connector type: Copper Pigtail, Cable technology: Passive Cable
  Link length support: 3m(active-copper)
  Capabilities: [Tx-Disable]
  Manufactured date: Tue Jan 16 00:00:00 GMT +0000 2024
  Tx-Disable(HW pin): False
```

Example for detailed view for all transceiver interfaces.

```
supervisor@rtbrick: op> show transceiver detail
Interface name: ifp-0/1/0
  Transceiver name: 25GBASE-CR, Vendor name: FS, Module type: SFP
  Serial number: S2108224424-2, Part number: S28-PC015, Material number: n/a
  Connector type: Copper Pigtail, Cable technology: Passive Cable
  Link length support: 3m(active-copper)  Manufactured date: Thu Sep 02 00:00:00 GMT
+0000 2021
  Tx-Disable(HW pin): False

Interface name: ifp-0/1/2
  Transceiver name: 10GBASE-CR Passive, Vendor name: FS, Module type: SFP
  Serial number: F2230685229-1, Part number: SFPP-PC005, Material number: n/a
```

```
  Connector type: Copper Pigtail, Cable technology: Passive Cable
  Link length support: 1m(active-copper)
  Capabilities: [Tx-Disable]
  Manufactured date: Wed Oct 18 00:00:00 GMT +0000 2023
  Tx-Disable(HW pin): False

Interface name: ifp-0/1/3
  Transceiver name: 10GBASE-CR Passive, Vendor name: FS, Module type: SFP
  Serial number: F2230685246-2, Part number: SFPP-PC005, Material number: n/a
  Connector type: Copper Pigtail, Cable technology: Passive Cable
  Link length support: 1m(active-copper)
  Capabilities: [Tx-Disable]
  Manufactured date: Wed Oct 18 00:00:00 GMT +0000 2023
  Tx-Disable(HW pin): False

Interface name: ifp-0/1/5
  Transceiver name: 10GBASE-CR Passive, Vendor name: FS, Module type: SFP
  Serial number: S2108213177-2, Part number: SFPP-PC01, Material number: n/a
  Connector type: Copper Pigtail, Cable technology: Passive Cable
  Link length support: 1m(active-copper)
  Capabilities: [Tx-Disable]
  Manufactured date: Thu Sep 02 00:00:00 GMT +0000 2021
  Tx-Disable(HW pin): False

Interface name: ifp-0/1/6
  Transceiver name: 10GBASE-CR Passive, Vendor name: FS, Module type: SFP
  Serial number: F2011195747-1, Part number: SFPP-PC015, Material number: n/a
  Connector type: Copper Pigtail, Cable technology: Passive Cable
  Link length support: 2m(active-copper)
  Capabilities: [Tx-Disable]
  Manufactured date: Thu Jan 18 00:00:00 GMT +0000 2024
  Tx-Disable(HW pin): False

Interface name: ifp-0/1/7
  Transceiver name: 10GBASE-CR Passive, Vendor name: FS, Module type: SFP
  Serial number: F2011195730-1, Part number: SFPP-PC015, Material number: n/a
  Connector type: Copper Pigtail, Cable technology: Passive Cable
  Link length support: 2m(active-copper)
  Capabilities: [Tx-Disable]
  Manufactured date: Thu Jan 18 00:00:00 GMT +0000 2024
  Tx-Disable(HW pin): False

Interface name: ifp-0/1/8
  Transceiver name: 10GBASE-CR Passive, Vendor name: FS, Module type: SFP
  Serial number: F2011195729-2, Part number: SFPP-PC015, Material number: n/a
  Connector type: Copper Pigtail, Cable technology: Passive Cable
  Link length support: 2m(active-copper)
  Capabilities: [Tx-Disable]
  Manufactured date: Thu Jan 18 00:00:00 GMT +0000 2024
  Tx-Disable(HW pin): False
```

```
Interface name: ifp-0/1/10
  Transceiver name: 10GBASE-LRM, Vendor name: FS, Module type: SFP
  Serial number: F2031324533, Part number: SFP-10GLRM-31, Material number:
COUIAR1CAC10-2456-03
  Power class: 1(1.5W), Power class mode: LOW
  Connector type: LC (Lucent Connector), Cable technology: Unknown
  Link length support: 1000m(om1) 2000m(om2), Wavelength: 1310.00 nm
  Capabilities: [Tx-Disable, Diagnostic Monitoring, Internal Calibration, Receiver
Power Measurement(AVG)]
  Manufactured date: Tue Jan 26 00:00:00 GMT +0000 2021
  Tx-Disable(HW pin): False
  Temperature parameters:
    Temperature: 27.25 °C
    Temperature low warning: False, Threshold: 0.00 °C
    Temperature low alarm: False, Threshold: -5.00 °C
    Temperature high warning: False, Threshold: 70.00 °C
    Temperature high alarm: False, Threshold: 75.00 °C
  Voltage parameters:
    Voltage: 3.29 V
    Voltage low warning: False, Threshold: 3.13 V
    Voltage low alarm: False, Threshold: 3.00 V
    Voltage high warning: False, Threshold: 3.47 V
    Voltage high alarm: False, Threshold: 3.60 V
  Lane Id: 0
    Interrupt parameters:
      Tx disable: False
      Laser Rx loss of signal: False
    Laser bias current parameters:
      Laser bias current: 25.290001 mA
      Laser bias current low warning: False, Threshold: 10.00 mA
      Laser bias current low alarm: False, Threshold: 5.00 mA
      Laser bias current high warning: False, Threshold: 85.00 mA
      Laser bias current high alarm: False, Threshold: 90.00 mA
    Laser Tx power parameters:
      Laser Tx power: 0.594300 mW / -2.259943 dBm
      Laser Tx power low warning: False, Threshold: 0.15 mW / -8.20 dBm
      Laser Tx power low alarm: False, Threshold: 0.12 mW / -9.20 dBm
      Laser Tx power high warning: False, Threshold: 1.12 mW / 0.50 dBm
      Laser Tx power high alarm: False, Threshold: 1.41 mW / 1.50 dBm
    Laser Rx power parameters:
      Laser Rx power: 0.313300 mW / -5.040396 dBm
      Laser Rx power low warning: False, Threshold: 0.04 mW / -14.41 dBm
      Laser Rx power low alarm: False, Threshold: 0.03 mW / -15.41 dBm
      Laser Rx power high warning: False, Threshold: 0.79 mW / -1.00 dBm
      Laser Rx power high alarm: False, Threshold: 1.00 mW / 0.00 dBm
Interface name: ifp-0/1/12
  Transceiver name: 10GBASE-LR, Vendor name: FS, Module type: SFP
  Serial number: F2031893724, Part number: SFP-10GLR-31, Material number:
COUIA75CAA10-2457-02
  Power class: 1(1.5W), Power class mode: LOW
  Connector type: LC (Lucent Connector), Cable technology: Unknown
```

```
  Link length support: 10000m(100m) 10km, Wavelength: 1310.00 nm
  Capabilities: [Tx-Disable, Diagnostic Monitoring, Internal Calibration, Receiver
Power Measurement(AVG)]
  Manufactured date: Tue Sep 01 00:00:00 GMT +0000 2020
  Tx-Disable(HW pin): False
  Temperature parameters:
    Temperature: 24.04 °C
    Temperature low warning: False, Threshold: 0.00 °C
    Temperature low alarm: False, Threshold: -5.00 °C
    Temperature high warning: False, Threshold: 70.00 °C
    Temperature high alarm: False, Threshold: 75.00 °C
  Voltage parameters:
    Voltage: 3.28 V
    Voltage low warning: False, Threshold: 3.13 V
    Voltage low alarm: False, Threshold: 3.00 V
    Voltage high warning: False, Threshold: 3.47 V
    Voltage high alarm: False, Threshold: 3.60 V
  Lane Id: 0
    Interrupt parameters:
      Tx disable: False
      Laser Rx loss of signal: False
    Laser bias current parameters:
      Laser bias current: 29.327999 mA
      Laser bias current low warning: False, Threshold: 10.00 mA
      Laser bias current low alarm: False, Threshold: 5.00 mA
      Laser bias current high warning: False, Threshold: 85.00 mA
      Laser bias current high alarm: False, Threshold: 90.00 mA
    Laser Tx power parameters:
      Laser Tx power: 0.515600 mW / -2.876871 dBm
      Laser Tx power low warning: False, Threshold: 0.15 mW / -8.20 dBm
      Laser Tx power low alarm: False, Threshold: 0.12 mW / -9.20 dBm
      Laser Tx power high warning: False, Threshold: 1.12 mW / 0.50 dBm
      Laser Tx power high alarm: False, Threshold: 1.41 mW / 1.50 dBm
    Laser Rx power parameters:
      Laser Rx power: 0.362300 mW / -4.409317 dBm
      Laser Rx power low warning: False, Threshold: 0.04 mW / -14.40 dBm
      Laser Rx power low alarm: False, Threshold: 0.03 mW / -15.41 dBm
      Laser Rx power high warning: False, Threshold: 0.79 mW / -1.00 dBm
      Laser Rx power high alarm: False, Threshold: 1.00 mW / 0.00 dBm
Interface name: ifp-0/1/14
  Transceiver name: 10GBASE-LR, Vendor name: FS, Module type: SFP
  Serial number: F2031893729, Part number: SFP-10GLR-31, Material number:
COUIA75CAA10-2457-02
  Power class: 1(1.5W), Power class mode: LOW
  Connector type: LC (Lucent Connector), Cable technology: Unknown
  Link length support: 10000m(100m) 10km, Wavelength: 1310.00 nm
  Capabilities: [Tx-Disable, Diagnostic Monitoring, Internal Calibration, Receiver
Power Measurement(AVG)]
  Manufactured date: Tue Sep 01 00:00:00 GMT +0000 2020
  Tx-Disable(HW pin): False
  Temperature parameters:
```

```
    Temperature: 24.26 °C
    Temperature low warning: False, Threshold: 0.00 °C
    Temperature low alarm: False, Threshold: -5.00 °C
    Temperature high warning: False, Threshold: 70.00 °C
    Temperature high alarm: False, Threshold: 75.00 °C
  Voltage parameters:
    Voltage: 3.27 V
    Voltage low warning: False, Threshold: 3.13 V
    Voltage low alarm: False, Threshold: 3.00 V
    Voltage high warning: False, Threshold: 3.47 V
    Voltage high alarm: False, Threshold: 3.60 V
  Lane Id: 0
    Interrupt parameters:
      Tx disable: False
      Laser Rx loss of signal: False
    Laser bias current parameters:
      Laser bias current: 29.186001 mA
      Laser bias current low warning: False, Threshold: 10.00 mA
      Laser bias current low alarm: False, Threshold: 5.00 mA
      Laser bias current high warning: False, Threshold: 85.00 mA
      Laser bias current high alarm: False, Threshold: 90.00 mA
    Laser Tx power parameters:
      Laser Tx power: 0.526800 mW / -2.783542 dBm
      Laser Tx power low warning: False, Threshold: 0.15 mW / -8.20 dBm
      Laser Tx power low alarm: False, Threshold: 0.12 mW / -9.20 dBm
      Laser Tx power high warning: False, Threshold: 1.12 mW / 0.50 dBm
      Laser Tx power high alarm: False, Threshold: 1.41 mW / 1.50 dBm
    Laser Rx power parameters:
      Laser Rx power: 0.335800 mW / -4.739193 dBm
      Laser Rx power low warning: False, Threshold: 0.04 mW / -14.40 dBm
      Laser Rx power low alarm: False, Threshold: 0.03 mW / -15.41 dBm
      Laser Rx power high warning: False, Threshold: 0.79 mW / -1.00 dBm
      Laser Rx power high alarm: False, Threshold: 1.00 mW / 0.00 dBm
Interface name: ifp-0/1/16
  Transceiver name: 25GBASE-SR, Vendor name: FS, Module type: SFP
  Serial number: G2006362714, Part number: SFP28-25GLR-31, Material number: n/a
  Power class: 1(1.5W), Power class mode: LOW
  Connector type: LC (Lucent Connector), Cable technology: Unknown
  Link length support: 10km, Wavelength: 1310.00 nm
  Capabilities: [Tx-Disable, Diagnostic Monitoring, Internal Calibration, Receiver
Power Measurement(AVG)]
  Manufactured date: Mon Aug 31 00:00:00 GMT +0000 2020
  Tx-Disable(HW pin): False
  Temperature parameters:
    Temperature: 28.44 °C
    Temperature low warning: False, Threshold: -45.00 °C
    Temperature low alarm: False, Threshold: -50.00 °C
    Temperature high warning: False, Threshold: 90.00 °C
    Temperature high alarm: False, Threshold: 95.00 °C
  Voltage parameters:
    Voltage: 3.29 V
```

```
    Voltage low warning: False, Threshold: 3.10 V
    Voltage low alarm: False, Threshold: 3.00 V
    Voltage high warning: False, Threshold: 3.50 V
    Voltage high alarm: False, Threshold: 3.60 V
  Lane Id: 0
    Interrupt parameters:
      Tx disable: False
      Laser Rx loss of signal: False
    Laser bias current parameters:
      Laser bias current: 38.625999 mA
      Laser bias current low warning: False, Threshold: 20.00 mA
      Laser bias current low alarm: False, Threshold: 10.00 mA
      Laser bias current high warning: False, Threshold: 80.00 mA
      Laser bias current high alarm: False, Threshold: 100.00 mA
    Laser Tx power parameters:
      Laser Tx power: 0.952700 mW / -0.210438 dBm
      Laser Tx power low warning: False, Threshold: 0.35 mW / -4.50 dBm
      Laser Tx power low alarm: False, Threshold: 0.28 mW / -5.50 dBm
      Laser Tx power high warning: False, Threshold: 2.24 mW / 3.50 dBm
      Laser Tx power high alarm: False, Threshold: 2.82 mW / 4.50 dBm
    Laser Rx power parameters:
      Laser Rx power: 0.917700 mW / -0.372993 dBm
      Laser Rx power low warning: False, Threshold: 0.04 mW / -14.50 dBm
      Laser Rx power low alarm: False, Threshold: 0.03 mW / -15.50 dBm
      Laser Rx power high warning: False, Threshold: 1.78 mW / 2.50 dBm
      Laser Rx power high alarm: False, Threshold: 2.24 mW / 3.50 dBm
Interface name: ifp-0/1/20
  Transceiver name: 25GBASE-CR, Vendor name: FS, Module type: SFP
  Serial number: G2220178363-1, Part number: S28-PC02, Material number: n/a
  Connector type: Copper Pigtail, Cable technology: Passive Cable
  Link length support: 3m(active-copper)  Manufactured date: Fri Jul 15 00:00:00 GMT
+0000 2022
  Tx-Disable(HW pin): False

Interface name: ifp-0/1/21
  Transceiver name: 25GBASE-CR, Vendor name: FS, Module type: SFP
  Serial number: G2130095858-2, Part number: S28-PC015, Material number: n/a
  Connector type: Copper Pigtail, Cable technology: Passive Cable
  Link length support: 3m(active-copper)  Manufactured date: Wed Nov 03 00:00:00 GMT
+0000 2021
  Tx-Disable(HW pin): False

Interface name: ifp-0/1/22
  Transceiver name: 10GBASE-CR Passive, Vendor name: FS, Module type: SFP
  Serial number: C2401249812-2, Part number: SFPP-PC025, Material number: n/a
  Connector type: Copper Pigtail, Cable technology: Passive Cable
  Link length support: 3m(active-copper)
  Capabilities: [Tx-Disable]
  Manufactured date: Tue Jan 16 00:00:00 GMT +0000 2024
  Tx-Disable(HW pin): False
```

```
Interface name: ifp-0/1/24
  Transceiver name: 10GBASE-CR Passive, Vendor name: FS, Module type: SFP
  Serial number: S2005019019-1, Part number: SFPP-PC03, Material number: n/a
  Connector type: Copper Pigtail, Cable technology: Passive Cable
  Link length support: 3m(active-copper) 3.200000m(cable-length)
  Capabilities: [Tx-Disable]
  Manufactured date: Fri May 08 00:00:00 GMT +0000 2020
  Tx-Disable(HW pin): False

Interface name: ifp-0/1/25
  Transceiver name: 10GBASE-CR Passive, Vendor name: FS, Module type: SFP
  Serial number: S2005019012-2, Part number: SFPP-PC03, Material number: n/a
  Connector type: Copper Pigtail, Cable technology: Passive Cable
  Link length support: 3m(active-copper) 3.200000m(cable-length)
  Capabilities: [Tx-Disable]
  Manufactured date: Fri May 08 00:00:00 GMT +0000 2020
  Tx-Disable(HW pin): False

Interface name: ifp-0/1/26
  Transceiver name: 10GBASE-CR Passive, Vendor name: FS, Module type: SFP
  Serial number: S2104008809-2, Part number: SFPP-PC02, Material number: n/a
  Connector type: Copper Pigtail, Cable technology: Passive Cable
  Link length support: 3m(active-copper)
  Capabilities: [Tx-Disable]
  Manufactured date: Fri Apr 09 00:00:00 GMT +0000 2021
  Tx-Disable(HW pin): False

Interface name: ifp-0/1/32
  Transceiver name: 10GBASE-LR, Vendor name: FLEXOPTIX, Module type: SFP
  Serial number: F7B7CSA, Part number: P.1396.10.I, Material number: n/a
  Power class: 1(1.5W), Power class mode: LOW
  Connector type: LC (Lucent Connector), Cable technology: Unknown
  Link length support: 10000m(100m) 10km, Wavelength: 1310.00 nm
  Capabilities: [Tx-Disable, Diagnostic Monitoring, Internal Calibration, Receiver
Power Measurement(AVG)]
  Manufactured date: Fri Feb 28 00:00:00 GMT +0000 2025
  Tx-Disable(HW pin): False
  Temperature parameters:
    Temperature: 32.67 °C
    Temperature low warning: False, Threshold: -40.00 °C
    Temperature low alarm: False, Threshold: -45.00 °C
    Temperature high warning: False, Threshold: 95.00 °C
    Temperature high alarm: False, Threshold: 100.00 °C
  Voltage parameters:
    Voltage: 3.36 V
    Voltage low warning: False, Threshold: 3.05 V
    Voltage low alarm: False, Threshold: 3.00 V
    Voltage high warning: False, Threshold: 3.50 V
    Voltage high alarm: False, Threshold: 3.60 V
  Lane Id: 0
    Interrupt parameters:
```

```
      Tx disable: False
      Laser Rx loss of signal: False
    Laser bias current parameters:
      Laser bias current: 37.591999 mA
      Laser bias current low warning: False, Threshold: 3.00 mA
      Laser bias current low alarm: False, Threshold: 2.00 mA
      Laser bias current high warning: False, Threshold: 80.00 mA
      Laser bias current high alarm: False, Threshold: 90.00 mA
    Laser Tx power parameters:
      Laser Tx power: 0.621800 mW / -2.063493 dBm
      Laser Tx power low warning: False, Threshold: 0.25 mW / -6.00 dBm
      Laser Tx power low alarm: False, Threshold: 0.20 mW / -7.00 dBm
      Laser Tx power high warning: False, Threshold: 1.41 mW / 1.50 dBm
      Laser Tx power high alarm: False, Threshold: 1.78 mW / 2.50 dBm
    Laser Rx power parameters:
      Laser Rx power: 0.591400 mW / -2.281187 dBm
      Laser Rx power low warning: False, Threshold: 0.03 mW / -15.00 dBm
      Laser Rx power low alarm: False, Threshold: 0.03 mW / -16.00 dBm
      Laser Rx power high warning: False, Threshold: 1.12 mW / 0.50 dBm
      Laser Rx power high alarm: False, Threshold: 1.26 mW / 1.00 dBm
Interface name: ifp-0/1/33
  Transceiver name: 10GBASE-LR, Vendor name: FLEXOPTIX, Module type: SFP
  Serial number: F7B7CSB, Part number: P.1396.10.I, Material number: n/a
  Power class: 1(1.5W), Power class mode: LOW
  Connector type: LC (Lucent Connector), Cable technology: Unknown
  Link length support: 10000m(100m) 10km, Wavelength: 1310.00 nm
  Capabilities: [Tx-Disable, Diagnostic Monitoring, Internal Calibration, Receiver
Power Measurement(AVG)]
  Manufactured date: Fri Feb 28 00:00:00 GMT +0000 2025
  Tx-Disable(HW pin): False
  Temperature parameters:
    Temperature: 32.42 °C
    Temperature low warning: False, Threshold: -40.00 °C
    Temperature low alarm: False, Threshold: -45.00 °C
    Temperature high warning: False, Threshold: 95.00 °C
    Temperature high alarm: False, Threshold: 100.00 °C
  Voltage parameters:
    Voltage: 3.36 V
    Voltage low warning: False, Threshold: 3.05 V
    Voltage low alarm: False, Threshold: 3.00 V
    Voltage high warning: False, Threshold: 3.50 V
    Voltage high alarm: False, Threshold: 3.60 V
  Lane Id: 0
    Interrupt parameters:
      Tx disable: False
      Laser Rx loss of signal: False
    Laser bias current parameters:
      Laser bias current: 34.501999 mA
      Laser bias current low warning: False, Threshold: 3.00 mA
      Laser bias current low alarm: False, Threshold: 2.00 mA
      Laser bias current high warning: False, Threshold: 80.00 mA
```

```
      Laser bias current high alarm: False, Threshold: 90.00 mA
    Laser Tx power parameters:
      Laser Tx power: 0.619400 mW / -2.080288 dBm
      Laser Tx power low warning: False, Threshold: 0.25 mW / -6.00 dBm
      Laser Tx power low alarm: False, Threshold: 0.20 mW / -7.00 dBm
      Laser Tx power high warning: False, Threshold: 1.41 mW / 1.50 dBm
      Laser Tx power high alarm: False, Threshold: 1.78 mW / 2.50 dBm
    Laser Rx power parameters:
      Laser Rx power: 0.615000 mW / -2.111249 dBm
      Laser Rx power low warning: False, Threshold: 0.03 mW / -15.00 dBm
      Laser Rx power low alarm: False, Threshold: 0.03 mW / -16.00 dBm
      Laser Rx power high warning: False, Threshold: 1.12 mW / 0.50 dBm
      Laser Rx power high alarm: False, Threshold: 1.26 mW / 1.00 dBm
Interface name: ifp-0/1/40
  Transceiver name: 25GBASE-CR, Vendor name: FS, Module type: SFP
  Serial number: S2108224423-1, Part number: S28-PC015, Material number: n/a
  Connector type: Copper Pigtail, Cable technology: Passive Cable
  Link length support: 3m(active-copper)  Manufactured date: Thu Sep 02 00:00:00 GMT
+0000 2021
  Tx-Disable(HW pin): False

Interface name: ifp-0/1/41
  Transceiver name: 10GBASE-CR Passive, Vendor name: FS, Module type: SFP
  Serial number: C2401249814-1, Part number: SFPP-PC025, Material number: n/a
  Connector type: Copper Pigtail, Cable technology: Passive Cable
  Link length support: 3m(active-copper)
  Capabilities: [Tx-Disable]
  Manufactured date: Tue Jan 16 00:00:00 GMT +0000 2024
  Tx-Disable(HW pin): False

Interface name: ifp-0/1/42
  Transceiver name: 10GBASE-CR Passive, Vendor name: FS, Module type: SFP
  Serial number: C2401249818-1, Part number: SFPP-PC025, Material number: n/a
  Connector type: Copper Pigtail, Cable technology: Passive Cable
  Link length support: 3m(active-copper)
  Capabilities: [Tx-Disable]
  Manufactured date: Tue Jan 16 00:00:00 GMT +0000 2024
  Tx-Disable(HW pin): False

Interface name: ifp-0/1/46
  Transceiver name: 10GBASE-CR Passive, Vendor name: FS, Module type: SFP
  Serial number: G2440559815-1, Part number: SFP-10G-PC01, Material number: n/a
  Connector type: Copper Pigtail, Cable technology: Passive Cable
  Link length support: 1m(active-copper)  Manufactured date: Wed Feb 12 00:00:00 GMT
+0000 2025
  Tx-Disable(HW pin): False

Interface name: ifp-0/1/47
  Transceiver name: 10GBASE-CR Passive, Vendor name: FS, Module type: SFP
  Serial number: G2440559809-1, Part number: SFP-10G-PC01, Material number: n/a
  Connector type: Copper Pigtail, Cable technology: Passive Cable
```

```
  Link length support: 1m(active-copper)  Manufactured date: Wed Feb 12 00:00:00 GMT
+0000 2025
  Tx-Disable(HW pin): False

Interface name: ifp-0/1/48
  Transceiver name: 10GBASE-CR Passive, Vendor name: FS, Module type: SFP
  Serial number: F2011195752-1, Part number: SFPP-PC015, Material number: n/a
  Connector type: Copper Pigtail, Cable technology: Passive Cable
  Link length support: 2m(active-copper)
  Capabilities: [Tx-Disable]
  Manufactured date: Thu Jan 18 00:00:00 GMT +0000 2024
  Tx-Disable(HW pin): False

Interface name: ifp-0/1/49
  Transceiver name: 10GBASE-CR Passive, Vendor name: FiberStore, Module type: SFP
  Serial number: F1712015010-1, Part number: SFP-10G-DAC, Material number: n/a
  Connector type: Copper Pigtail, Cable technology: Passive Cable
  Link length support: 2m(active-copper)
  Capabilities: [Tx-Disable]
  Manufactured date: Thu Dec 21 00:00:00 GMT +0000 2017
  Tx-Disable(HW pin): False

Interface name: ifp-0/1/89
  Transceiver name: 25GAUI-C2M-AOC, Vendor name: FS, Module type: SFP
  Serial number: C2403495328-1, Part number: Q-4S28AO01, Material number: n/a
  Power class: 1(1.5W), Power class mode: LOW
  Connector type: No separable connector, Cable technology: Active Cable
  Link length support: 10m(om4), Wavelength: 0.00 nm
  Capabilities: [Diagnostic Monitoring, Internal Calibration, Receiver Power
Measurement(AVG)]
  Manufactured date: Thu Mar 28 00:00:00 GMT +0000 2024
  Tx-Disable(HW pin): False
  Temperature parameters:
    Temperature: 32.00 °C
    Temperature low warning: False, Threshold: 0.00 °C
    Temperature low alarm: False, Threshold: -10.00 °C
    Temperature high warning: False, Threshold: 70.00 °C
    Temperature high alarm: False, Threshold: 80.00 °C
  Voltage parameters:
    Voltage: 3.27 V
    Voltage low warning: False, Threshold: 3.13 V
    Voltage low alarm: False, Threshold: 2.97 V
    Voltage high warning: False, Threshold: 3.46 V
    Voltage high alarm: False, Threshold: 3.63 V
  Lane Id: 0
    Interrupt parameters:
      Tx disable: False
      Laser Rx loss of signal: False
    Laser bias current parameters:
      Laser bias current: 6.220000 mA
      Laser bias current low warning: False, Threshold: 2.00 mA
```

```
      Laser bias current low alarm: False, Threshold: 1.00 mA
      Laser bias current high warning: False, Threshold: 12.00 mA
      Laser bias current high alarm: False, Threshold: 15.00 mA
    Laser Tx power parameters:
      Laser Tx power: 1.465700 mW / 1.660451 dBm
      Laser Tx power low warning: False, Threshold: 0.25 mW / -6.00 dBm
      Laser Tx power low alarm: False, Threshold: 0.13 mW / -9.00 dBm
      Laser Tx power high warning: False, Threshold: 2.00 mW / 3.01 dBm
      Laser Tx power high alarm: False, Threshold: 2.24 mW / 3.50 dBm
    Laser Rx power parameters:
      Laser Rx power: 1.044500 mW / 0.189084 dBm
      Laser Rx power low warning: False, Threshold: 0.10 mW / -10.00 dBm
      Laser Rx power low alarm: False, Threshold: 0.05 mW / -13.00 dBm
      Laser Rx power high warning: False, Threshold: 2.00 mW / 3.01 dBm
      Laser Rx power high alarm: False, Threshold: 2.24 mW / 3.50 dBm
Interface name: ifp-0/1/91
  Transceiver name: 25GAUI-C2M-AOC, Vendor name: FS, Module type: SFP
  Serial number: C2403495328-4, Part number: Q-4S28AO01, Material number: n/a
  Power class: 1(1.5W), Power class mode: LOW
  Connector type: No separable connector, Cable technology: Active Cable
  Link length support: 10m(om4), Wavelength: 0.00 nm
  Capabilities: [Diagnostic Monitoring, Internal Calibration, Receiver Power
Measurement(AVG)]
  Manufactured date: Thu Mar 28 00:00:00 GMT +0000 2024
  Tx-Disable(HW pin): False
  Temperature parameters:
    Temperature: 32.00 °C
    Temperature low warning: False, Threshold: 0.00 °C
    Temperature low alarm: False, Threshold: -10.00 °C
    Temperature high warning: False, Threshold: 70.00 °C
    Temperature high alarm: False, Threshold: 80.00 °C
  Voltage parameters:
    Voltage: 3.27 V
    Voltage low warning: False, Threshold: 3.13 V
    Voltage low alarm: False, Threshold: 2.97 V
    Voltage high warning: False, Threshold: 3.46 V
    Voltage high alarm: False, Threshold: 3.63 V
  Lane Id: 0
    Interrupt parameters:
      Tx disable: False
      Laser Rx loss of signal: False
    Laser bias current parameters:
      Laser bias current: 6.226000 mA
      Laser bias current low warning: False, Threshold: 2.00 mA
      Laser bias current low alarm: False, Threshold: 1.00 mA
      Laser bias current high warning: False, Threshold: 12.00 mA
      Laser bias current high alarm: False, Threshold: 15.00 mA
    Laser Tx power parameters:
      Laser Tx power: 1.467200 mW / 1.664893 dBm
      Laser Tx power low warning: False, Threshold: 0.25 mW / -6.00 dBm
      Laser Tx power low alarm: False, Threshold: 0.13 mW / -9.00 dBm
```

```
      Laser Tx power high warning: False, Threshold: 2.00 mW / 3.01 dBm
      Laser Tx power high alarm: False, Threshold: 2.24 mW / 3.50 dBm
    Laser Rx power parameters:
      Laser Rx power: 1.106600 mW / 0.439907 dBm
      Laser Rx power low warning: False, Threshold: 0.10 mW / -10.00 dBm
      Laser Rx power low alarm: False, Threshold: 0.05 mW / -13.00 dBm
      Laser Rx power high warning: False, Threshold: 2.00 mW / 3.01 dBm
      Laser Rx power high alarm: False, Threshold: 2.24 mW / 3.50 dBm
Interface name: ifp-0/1/93
  Transceiver name: 25GAUI-C2M-AOC, Vendor name: FS, Module type: SFP
  Serial number: C2403495328-2, Part number: Q-4S28AO01, Material number: n/a
  Power class: 1(1.5W), Power class mode: LOW
  Connector type: No separable connector, Cable technology: Active Cable
  Link length support: 10m(om4), Wavelength: 0.00 nm
  Capabilities: [Diagnostic Monitoring, Internal Calibration, Receiver Power
Measurement(AVG)]
  Manufactured date: Thu Mar 28 00:00:00 GMT +0000 2024
  Tx-Disable(HW pin): False
  Temperature parameters:
    Temperature: 31.00 °C
    Temperature low warning: False, Threshold: 0.00 °C
    Temperature low alarm: False, Threshold: -10.00 °C
    Temperature high warning: False, Threshold: 70.00 °C
    Temperature high alarm: False, Threshold: 80.00 °C
  Voltage parameters:
    Voltage: 3.28 V
    Voltage low warning: False, Threshold: 3.13 V
    Voltage low alarm: False, Threshold: 2.97 V
    Voltage high warning: False, Threshold: 3.46 V
    Voltage high alarm: False, Threshold: 3.63 V
  Lane Id: 0
    Interrupt parameters:
      Tx disable: False
      Laser Rx loss of signal: False
    Laser bias current parameters:
      Laser bias current: 6.206000 mA
      Laser bias current low warning: False, Threshold: 2.00 mA
      Laser bias current low alarm: False, Threshold: 1.00 mA
      Laser bias current high warning: False, Threshold: 12.00 mA
      Laser bias current high alarm: False, Threshold: 15.00 mA
    Laser Tx power parameters:
      Laser Tx power: 1.456800 mW / 1.633999 dBm
      Laser Tx power low warning: False, Threshold: 0.25 mW / -6.00 dBm
      Laser Tx power low alarm: False, Threshold: 0.13 mW / -9.00 dBm
      Laser Tx power high warning: False, Threshold: 2.00 mW / 3.01 dBm
      Laser Tx power high alarm: False, Threshold: 2.24 mW / 3.50 dBm
    Laser Rx power parameters:
      Laser Rx power: 1.053600 mW / 0.226758 dBm
      Laser Rx power low warning: False, Threshold: 0.10 mW / -10.00 dBm
      Laser Rx power low alarm: False, Threshold: 0.05 mW / -13.00 dBm
      Laser Rx power high warning: False, Threshold: 2.00 mW / 3.01 dBm
```

```
      Laser Rx power high alarm: False, Threshold: 2.24 mW / 3.50 dBm
Interface name: ifp-0/1/95
  Transceiver name: 25GAUI-C2M-AOC, Vendor name: FS, Module type: SFP
  Serial number: C2403495328-3, Part number: Q-4S28AO01, Material number: n/a
  Power class: 1(1.5W), Power class mode: LOW
  Connector type: No separable connector, Cable technology: Active Cable
  Link length support: 10m(om4), Wavelength: 0.00 nm
  Capabilities: [Diagnostic Monitoring, Internal Calibration, Receiver Power
Measurement(AVG)]
  Manufactured date: Thu Mar 28 00:00:00 GMT +0000 2024
  Tx-Disable(HW pin): False
  Temperature parameters:
    Temperature: 31.00 °C
    Temperature low warning: False, Threshold: 0.00 °C
    Temperature low alarm: False, Threshold: -10.00 °C
    Temperature high warning: False, Threshold: 70.00 °C
    Temperature high alarm: False, Threshold: 80.00 °C
  Voltage parameters:
    Voltage: 3.28 V
    Voltage low warning: False, Threshold: 3.13 V
    Voltage low alarm: False, Threshold: 2.97 V
    Voltage high warning: False, Threshold: 3.46 V
    Voltage high alarm: False, Threshold: 3.63 V
  Lane Id: 0
    Interrupt parameters:
      Tx disable: False
      Laser Rx loss of signal: False
    Laser bias current parameters:
      Laser bias current: 6.206000 mA
      Laser bias current low warning: False, Threshold: 2.00 mA
      Laser bias current low alarm: False, Threshold: 1.00 mA
      Laser bias current high warning: False, Threshold: 12.00 mA
      Laser bias current high alarm: False, Threshold: 15.00 mA
    Laser Tx power parameters:
      Laser Tx power: 1.463200 mW / 1.653037 dBm
      Laser Tx power low warning: False, Threshold: 0.25 mW / -6.00 dBm
      Laser Tx power low alarm: False, Threshold: 0.13 mW / -9.00 dBm
      Laser Tx power high warning: False, Threshold: 2.00 mW / 3.01 dBm
      Laser Tx power high alarm: False, Threshold: 2.24 mW / 3.50 dBm
    Laser Rx power parameters:
      Laser Rx power: 1.067500 mW / 0.283679 dBm
      Laser Rx power low warning: False, Threshold: 0.10 mW / -10.00 dBm
      Laser Rx power low alarm: False, Threshold: 0.05 mW / -13.00 dBm
      Laser Rx power high warning: False, Threshold: 2.00 mW / 3.01 dBm
      Laser Rx power high alarm: False, Threshold: 2.24 mW / 3.50 dBm
Interface name: ifp-0/1/96
  Transceiver name: 100GBASE-CR4, Vendor name: Fiberstore, Module type: QSFP28
  Serial number: F1800034265-1, Part number: Q28-PC01, Material number: n/a
  Connector type: No separable connector
  Link length support: 1m(om4)  Manufactured date: Tue Jan 16 00:00:00 GMT +0000 2018
  Lane Id: 0
```

```
  Lane Id: 1
  Lane Id: 2
  Lane Id: 3
Interface name: ifp-0/1/97
  Transceiver name: 100GBASE-CR4, Vendor name: FS, Module type: QSFP28
  Serial number: G2110248567-2, Part number: Q28-PC005, Material number: n/a
  Connector type: No separable connector
  Link length support: 1m(om4)  Manufactured date: Sat May 08 00:00:00 GMT +0000 2021
  Lane Id: 0
  Lane Id: 1
  Lane Id: 2
  Lane Id: 3
Interface name: ifp-0/1/100
  Transceiver name: 100G-AOC, Vendor name: FS, Module type: QSFP28
  Serial number: C2403495328, Part number: Q-4S28AO01, Material number:
....................
  Power class: 4(3.5W), Power class mode: HIGH
  Connector type: No separable connector
  Link length support: 2m(om4), Wavelength: 0.00 nm
  Capabilities: [Tx-Disable, Diagnostic Monitoring, Transmitter Power Measurement,
Receiver Power Measurement(AVG)]
  Manufactured date: Tue Mar 19 00:00:00 GMT +0000 2024
  Temperature parameters:
    Temperature: 27.78 °C
    Temperature low warning: False, Threshold: -3.00 °C
    Temperature low alarm: False, Threshold: -5.00 °C
    Temperature high warning: False, Threshold: 73.00 °C
    Temperature high alarm: False, Threshold: 75.00 °C
  Voltage parameters:
    Voltage: 3.28 V
    Voltage low warning: False, Threshold: 3.13 V
    Voltage low alarm: False, Threshold: 2.97 V
    Voltage high warning: False, Threshold: 3.46 V
    Voltage high alarm: False, Threshold: 3.63 V
  Lane Id: 0
    Interrupt parameters:
      Tx disable: False, Laser Tx loss of signal: False, Laser Tx loss of lock: False
      Laser Rx loss of signal: False, Laser Rx loss of lock: False
    Laser bias current parameters:
      Laser bias current: 7.018000 mA
      Laser bias current low warning: False, Threshold: 2.00 mA
      Laser bias current low alarm: False, Threshold: 1.00 mA
      Laser bias current high warning: False, Threshold: 12.00 mA
      Laser bias current high alarm: False, Threshold: 15.00 mA
    Laser Tx power parameters:
      Laser Tx power: 1.096300 mW / 0.399294 dBm
      Laser Tx power low warning: False, Threshold: 0.14 mW / -8.40 dBm
      Laser Tx power low alarm: False, Threshold: 0.09 mW / -10.40 dBm
      Laser Tx power high warning: False, Threshold: 1.74 mW / 2.40 dBm
      Laser Tx power high alarm: False, Threshold: 2.19 mW / 3.40 dBm
    Laser Rx power parameters:
```

```
      Laser Rx power: 0.727700 mW / -1.380476 dBm
      Laser Rx power low warning: False, Threshold: 0.09 mW / -10.30 dBm
      Laser Rx power low alarm: False, Threshold: 0.06 mW / -12.30 dBm
      Laser Rx power high warning: False, Threshold: 1.74 mW / 2.40 dBm
      Laser Rx power high alarm: False, Threshold: 2.19 mW / 3.40 dBm
  Lane Id: 1
    Interrupt parameters:
      Tx disable: False, Laser Tx loss of signal: False, Laser Tx loss of lock: False
      Laser Rx loss of signal: False, Laser Rx loss of lock: False
    Laser bias current parameters:
      Laser bias current: 7.012000 mA
      Laser bias current low warning: False, Threshold: 2.00 mA
      Laser bias current low alarm: False, Threshold: 1.00 mA
      Laser bias current high warning: False, Threshold: 12.00 mA
      Laser bias current high alarm: False, Threshold: 15.00 mA
    Laser Tx power parameters:
      Laser Tx power: 1.135000 mW / 0.549959 dBm
      Laser Tx power low warning: False, Threshold: 0.14 mW / -8.40 dBm
      Laser Tx power low alarm: False, Threshold: 0.09 mW / -10.40 dBm
      Laser Tx power high warning: False, Threshold: 1.74 mW / 2.40 dBm
      Laser Tx power high alarm: False, Threshold: 2.19 mW / 3.40 dBm
    Laser Rx power parameters:
      Laser Rx power: 0.776800 mW / -1.096908 dBm
      Laser Rx power low warning: False, Threshold: 0.09 mW / -10.30 dBm
      Laser Rx power low alarm: False, Threshold: 0.06 mW / -12.30 dBm
      Laser Rx power high warning: False, Threshold: 1.74 mW / 2.40 dBm
      Laser Rx power high alarm: False, Threshold: 2.19 mW / 3.40 dBm
  Lane Id: 2
    Interrupt parameters:
      Tx disable: False, Laser Tx loss of signal: False, Laser Tx loss of lock: False
      Laser Rx loss of signal: False, Laser Rx loss of lock: False
    Laser bias current parameters:
      Laser bias current: 7.004000 mA
      Laser bias current low warning: False, Threshold: 2.00 mA
      Laser bias current low alarm: False, Threshold: 1.00 mA
      Laser bias current high warning: False, Threshold: 12.00 mA
      Laser bias current high alarm: False, Threshold: 15.00 mA
    Laser Tx power parameters:
      Laser Tx power: 1.171800 mW / 0.688535 dBm
      Laser Tx power low warning: False, Threshold: 0.14 mW / -8.40 dBm
      Laser Tx power low alarm: False, Threshold: 0.09 mW / -10.40 dBm
      Laser Tx power high warning: False, Threshold: 1.74 mW / 2.40 dBm
      Laser Tx power high alarm: False, Threshold: 2.19 mW / 3.40 dBm
    Laser Rx power parameters:
      Laser Rx power: 1.146800 mW / 0.594877 dBm
      Laser Rx power low warning: False, Threshold: 0.09 mW / -10.30 dBm
      Laser Rx power low alarm: False, Threshold: 0.06 mW / -12.30 dBm
      Laser Rx power high warning: False, Threshold: 1.74 mW / 2.40 dBm
      Laser Rx power high alarm: False, Threshold: 2.19 mW / 3.40 dBm
  Lane Id: 3
    Interrupt parameters:
```

```
      Tx disable: False, Laser Tx loss of signal: False, Laser Tx loss of lock: False
      Laser Rx loss of signal: False, Laser Rx loss of lock: False
    Laser bias current parameters:
      Laser bias current: 6.998000 mA
      Laser bias current low warning: False, Threshold: 2.00 mA
      Laser bias current low alarm: False, Threshold: 1.00 mA
      Laser bias current high warning: False, Threshold: 12.00 mA
      Laser bias current high alarm: False, Threshold: 15.00 mA
    Laser Tx power parameters:
      Laser Tx power: 1.086400 mW / 0.359898 dBm
      Laser Tx power low warning: False, Threshold: 0.14 mW / -8.40 dBm
      Laser Tx power low alarm: False, Threshold: 0.09 mW / -10.40 dBm
      Laser Tx power high warning: False, Threshold: 1.74 mW / 2.40 dBm
      Laser Tx power high alarm: False, Threshold: 2.19 mW / 3.40 dBm
    Laser Rx power parameters:
      Laser Rx power: 1.054700 mW / 0.231289 dBm
      Laser Rx power low warning: False, Threshold: 0.09 mW / -10.30 dBm
      Laser Rx power low alarm: False, Threshold: 0.06 mW / -12.30 dBm
      Laser Rx power high warning: False, Threshold: 1.74 mW / 2.40 dBm
      Laser Rx power high alarm: False, Threshold: 2.19 mW / 3.40 dBm
Interface name: ifp-0/1/101
  Transceiver name: 100GBASE-CR4, Vendor name: FS, Module type: QSFP28
  Serial number: G2006417082-1, Part number: Q28-PC02, Material number: n/a
  Connector type: No separable connector
  Link length support: 3m(om4)  Manufactured date: Fri Oct 09 00:00:00 GMT +0000 2020
  Lane Id: 0
  Lane Id: 1
  Lane Id: 2
  Lane Id: 3
```

#### Resetting the Optical Transceiver Manually

The `reset transceiver` operational command allows users to manually reset an optical transceiver on a specified interface.

**Syntax:**

**reset transceiver interface** <interface>

| Attribute | Description |
|---|---|
| `<interface>` | Name of the physical interface. |

The following example shows how to manually reset the optical transceiver on the specified interface, 'ifp-0/1/64'. In this command:

- For QSFP transceivers, a hardware reset signal is sent via I2C to reset the transceiver hardware.
- Port disable and enable are performed.

```
supervisor@rtbrick: op> reset transceiver interface ifp-0/1/64
  Transceiver ifp-0/1/64 hardware reset signal sent via I2C
  Initiating state transition (down/up) on port ifp-0/1/64 (bcm port 65)
  Port ifp-0/1/64 administrative state set to down
  Port ifp-0/1/64 (bcm port 65) administrative state set to up
```

**Disk Details**

This command displays disk information.

**Syntax:**

**show disk** <option>

| Option | Description |
|---|---|
| `summary` | Displays the disk information. |
| `usage` | Displays the disk usage information. |

Example: Summary of disks and their partitions.

```
supervisor@rtbrick: op> show disk summary
sda
  Size    : 119.2G
  Vendor  : ATA
  Model   : SH250-M280 128GB
  Partitions
    Name               Size        Mount Point
    sda1               256M        n/a
    sda2               128M        n/a
    sda3               1M          n/a
    sda4               4G          /var/config
    sda5               24.4G       /var/log
    sda6               56.9G       /var/crash
    sda7               16G         /usr/local/bin/bcm/t
    sda8               16G         /usr/local/bin/bcm/t
```

Example: Summary of disk usage information.

```
supervisor@rtbrick: op> show disk usage
Filesystem               Type             Size         Used         Available
Mountpoint          Usage %
none                     tmpfs            492 KiB      4 KiB        488 KiB      /dev
0.81
tmpfs                    tmpfs            15.51 GiB    8.15 MiB     15.5 GiB     /run
0.05
```

```
tmpfs                    tmpfs            6 GiB        0 bytes      6 GiB        /shm
0.0
udev                     devtmpfs         15.47 GiB    0 bytes      15.47 GiB
/dev/mem            0.0
tmpfs                    tmpfs            15.51 GiB    9.53 MiB     15.5 GiB
/dev/shm            0.06
tmpfs                    tmpfs            5 MiB        0 bytes      5 MiB
/run/lock           0.0
/dev/sda5                ext4             23.82 GiB    17.77 MiB    22.57 GiB
/var/log            0.08
/dev/sda7                ext4             15.58 GiB    4.56 GiB     10.21 GiB
/platform           30.87
/dev/sda6                ext4             55.67 GiB    48 KiB       52.82 GiB
/var/crash          0.0
/dev/sda4                ext4             3.86 GiB     104 KiB      3.64 GiB
/var/config         0.0
tmpfs                    tmpfs            3.1 GiB      1.25 MiB     3.1 GiB
/var/run-ext/onl/r  0.04
/var/cache/rtbrick/imag  overlay          15.58 GiB    4.56 GiB     10.21 GiB    /
30.87
```

**Platform Details**

This command displays platform information.

**Syntax:**

```
show platform
```

Example: Platform information

```
supervisor@rtbrick: op> show platform
x86_64-ufispace_s9600_102xc-r0
  Role                 : multiservice-edge
  Platform             : q2c(BCM88820_A1)
  External Processor   : OP2(model: X register-value: 0x69A)
  Vendor               : Ufi Space
  Manufacturer         : Ufi Space
  Manufacture date     : 09/23/2023 23:44:31
  MAC address          : 5C:07:58:79:0A:00
  Part number          : S9600-102XC-RBZB
  Serial number        : WYA1D9VF00002B1
  Product name         : S9600-102XC-R
  ONIE version         : 2022.02v04
  Label revision       : N/A
  Diag version         : 0.0.2
  Country code         : VN
  Device version       : 0
```

### Hardware Resource Usage Information

The command `show hardware resource monitor` displays the hardware consumption information for the various resources.

**Syntax:**

**show hardware resource monitor**

Example: Hardware resource monitoring information

```
supervisor@rtbrick: op> show hardware resource monitor
  Resource Type                                          Consumed Estimated
  IPV4_UNICAST_PRIVATE_LPM_FORWARD                              0   1704960
  IPV4_UNICAST_PRIVATE_LPM_FORWARD_2                            0    189440
  IPV6_UNICAST_PRIVATE_LPM_FORWARD                              0    852480
  IPV6_UNICAST_PRIVATE_LPM_FORWARD_2                            0     94720
  IPV4_MULTICAST_PRIVATE_LPM_FORWARD                            0    852480
  IPV4_MULTICAST_TCAM_FORWARD                                   0     75776
  MPLS_TERMINATION_SINGLE_LABEL_DB                              3    125171
  KBP_IPV4_UNICAST_PRIVATE_LPM_FORWARD                          0   8563370
  KBP_IPV6_UNICAST_PRIVATE_LPM_FORWARD                          0   6422528
  IN_AC_C_C_VLAN_DB                                             0    141390
  IN_AC_C_VLAN_DB                                               7    141390
  IN_AC_UNTAGGED_DB                                             0    141390
  IN_AC_S_C_VLAN_DB                                             0    141390
  IN_AC_S_VLAN_DB                                               0    141390
  MPLS_FWD                                                      0    186777
  MULTICAST_MCDB                                                1    262142
  L3_RIF                                                        9     40959
  EEDB_MPLS_TUNNEL                                              0     81920
  EEDB_PWE                                                      0     81920
  EEDB_L2TP                                                     0     81920
  L2TPV2_DATA_MESSAGE_TT                                        0    125171
  EEDB_PPPOE                                                    0     81920
  EEDB_ARP                                                      9     81920
  EEDB_IPV4_TUNNEL                                              0     81920
  EEDB_IPV6_TUNNEL                                              0     81920
  IPV4_TT_P2P_EM_BASIC                                          0     57834
  IPV6_TT_P2P_EM_BASIC                                          0     57834
  PPPOE_O_ETH_TUNNEL_FULL_SA                                    0     57834
  IN_LIF_FORMAT_PWE                                             0     98304
  ECMP_GROUP                                                    0     40960
  QOS_VOQ_CONNECTOR_BUNDLES                                    41     32000
  QOS_UC_QUEUE_BUNDLES                                        161     32000
  QOS_MC_QUEUE_BUNDLES                                          1     32000
  QOS_CL_ELEMENTS                                               0     30000
  QOS_FQ_ELEMENTS                                               0     30000
  QOS_HR_ELEMENTS                                             240     30000
```

```
  QOS_SCH_SHARED_SHAPER_DUAL                                  0     30000
  QOS_SCH_SHARED_SHAPER_QUAD                                  0     30000
  QOS_SCH_SHARED_SHAPER_OCTA                                  0     30000
  MFV4_EXTERNAL                                               0      2048
  MFV6_EXTERNAL                                               0      1024
  L3V4_EXTERNAL                                               3      1024
  L3V6_EXTERNAL                                               6      1024
  POLICER_EXTERNAL                                            0      4096
  KBP_IPV4_UNICAST_PRIVATE_LPM_RPF                            0   7315865
  KBP_IPV6_UNICAST_PRIVATE_LPM_RPF                            0   4912878
  QOS_EGRESS_PROFILE_TO_INTERNAL_MAP                         11       256
  QOS_INGRESS_PHB_PROFILE_TO_INTERNAL_MAP                     9      9216
  Cross Connect                                               0     20000
```

Example output of the `show hardware resource monitor` command for the UfiSpace S9500-22XST multiservice edge platform. It includes information on IPv4, IPv6, MPLS, and L2VPN cross-connect resource utilization.

```
supervisor@rtbrick.net: cfg> show hardware resource monitor
  Resource Type                                          Consumed Estimated
  Cross Connect                                              4000     20000
  Double tagged LIF                                             0     65000
  ECMP id                                                       0      4095
  Egress Failover id                                            0     32767
  FEC Failover id                                               0     32765
  FECs for global use                                        2009     61440
  Field entry id                                                2     65538
  Field extraction entry id                                    15        64
  IPv4 UC KAPS                                                  0     50000
  IPv4 UC LEM                                                   1     50000
  IPv6 UC Route                                                 3     20000
  Ingress Failover id                                           0     32767
  Local Common InLif                                         5011     65536
  Local OutLif                                              10033    163840
  Local Wide InLif                                           3341     43690
  MPLS LSR LEM                                               2000     20000
  MPLS SEMA                                                     0     20000
  MPLS SEMB                                                     0     20000
  Number of meters in processor A                               0     32768
  Number of meters in processor B                               0     32768
  Profiles for PON use                                          0        16
  QOS EGRESS DSCP/EXP MARKING PROFILE IDs                       0        16
  QOS EGRESS L2 I TAG PROFILE IDs                               0         1
  QOS EGRESS MPLS PHP QOS IDs                                   0         3
  QOS EGRESS REMARK QOS IDs                                     3        15
  QOS INGRESS COS OPCODE IDs                                    0         7
  QOS INGRESS LABEL MAP ID                                      0         1
  QOS INGRESS LIF/COS IDs                                       1        63
  QOS INGRESS PCP PROFILE IDs                                   0        15
  SW handles of policer                                         0         7
```

```
  Single tagged LIF                                    2004     65000
  Trill virtual nickname                                  0         2
  Untag LIF                                               0     65000
  VLAN translation egress usage                          14       512
  VLAN translation ingress usage                         11        64
  VSIs for MSTP                                          16     28672
  VSIs for TB VLANS                                       1      4096
```

**Error Message Indicator for Transceiver**

An error indicator in the output of the `show transceiver <ifp-name>` and `show interface <ifp-name>` commands. It flags an error message for broken or faulty transceivers or when the transceiver EEPROM is corrupted or cannot be read. It allows for quick fault identification without the need for additional diagnostic steps.

The following is a show command output displays an indicator for the specified transceiver 'ifp-0/1/10'.

```
supervisor@rtbrick.net: cfg> show transceiver ifp-0/1/10
ifp-0/1/10
  Transceiver name: n/a, Vendor name: n/a, Module type: Unknown(Transceiver-EEPROM
corruption/read-failure), Serial number: n/a
  Part number: n/a, Material number: n/a
  Power class: n/a
  Connector type: n/a, Link length support: n/a  Manufactured date: n/a
```

# 10.3. SNMP

## 10.3.1. SNMP Overview

SNMP (Simple Network Management Protocol) provides a network monitoring mechanism that collects state information from various network devices and components. The protocol enables you to monitor the RBFS network and detect network faults on remote devices.

SNMP can monitor interfaces, CPU usage, temperature of the device, bandwidth usage, and so on. For example, if an interface goes down on one of the devices, SNMP can quickly alert the change.

**Understanding RBFS SNMP Implementation**

The RBFS SNMP implementation allows retrieving system state information using the Protocol Data Unit (PDU) from various network components. SNMP defines system objects in Management Information Bases (MIBs), where each object forms a Protocol Data Unit (PDU).

A device that is SNMP enabled is known as SNMP agent. SNMP agent collects information from various devices and components and stores the data within MIB. An SNMP agent includes several objects such as interfaces and routing tables which can be interacted with. Every object has a unique identifier which is known as OID (Object Identifier). OIDs provide a unique identity for managed objects in an MIB hierarchy.

RBFS implements the SNMP daemon (`snmpd`) that maps the operational state API to SNMP MIBs for retrieving system state data.

### SNMP Operations

SNMP allows performing various operations that include GET for retrieving data, SET for modifying data, TRAP for notifying an event and so on. These operations provide management access to the MIB hierarchy.

1. **GET**: The SNMP GET operation retrieves data from the managed device's MIB.
2. **GETNEXT**: The GETNEXT operation retrieves data for the next object from the tree of objects on the device.
3. **SET**: The SNMP SET operation allows modifying data for a device.
4. **TRAP**: Event notification that is not requested.

> RBFS does not currently support the SNMP SET operation.

### Supported SNMP MIBs

Management Information Base (MIB) is a collection of data that is organized hierarchically. You can define an MIB mapping for the supported MIBs. This MIB mapping provides a way to retrieve the required state information using the Operational State API or any other RBFS API or Prometheus.

Currently, RBFS supports two SNMP MIBs: the Interface MIB and the Host Resources MIBs.

- **Host Resources MIB**: Provides information about system resources on the host machine, such as CPU, memory, and storage. For details, see https://www.net-snmp.org/docs/mibs/HOST-RESOURCES-MIB.txt.
- **Interfaces MIB**: Provides information about the status and statistics of network interfaces. For details, see https://www.net-snmp.org/docs/mibs/IF-MIB.txt.
- **System MIB**: Supplies general device information, including system name, hardware details, and the installed network operating system version. https://www.net-snmp.org/docs/mibs/RFC1213-MIB.txt.

### Supported SNMP Versions

RBFS supports the SNMP version 2c and SNMP version 3. SNMP version 3 provides support for authentication and encryption and the data can be accessed after capture only by authorized users. You must choose either version 2c or version 3 before configuring other functionalities.

### Information about SNMP v2c and v3

SNMP v2c allows access control through Communities, but the community information is not protected. Anyone with the community name can access all the information available for that community.

SNMP v3 provides a higher level of security using authentication and privacy protocols. RBFS supports the user-based security model defined in RFC 3414.

The user credentials verify the authenticity and integrity of the message by adding message authentication codes (MACs) to the SNMP packet. The privacy protocol allows for the encryption of the transmitted data.

**Supported Platforms**

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

## 10.3.2. SNMP Configuration

By default, SNMP is not enabled. To enable SNMP, you must complete the SNMP configurations.

RBFS supports SNMP 2c and 3 versions. You must first configure the desired version before configuring other functionalities. RBFS does not support running both the SNMP 2c and 3 versions at the same time.

The RBFS CLI displays all options regardless of the selected version. Even though functionalities such as Community can be configured for both of the versions, it works only in SNMP 2c version. You can define 'user profiles' on SNMP v3 and 'Community' in version 2c. Similarly, Engine ID can be defined in SNMP version 3.

**Configuration Hierarchy**

The diagram illustrates the SNMP configuration hierarchy. All SNMP configurations are performed within an instance.

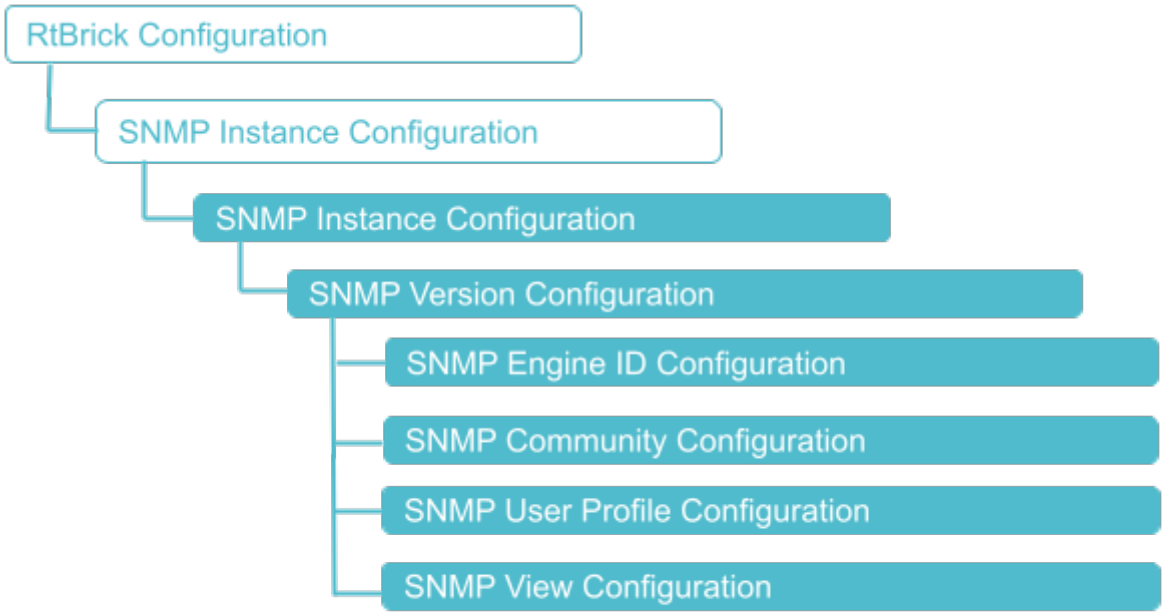



*Figure 161. SNMP Configuration Hierarchy*

### Configuration Syntax and Commands

The following sections describe the SNMP configuration syntax and commands.

#### SNMP Instance Configuration

At this instance configuration hierarchy, you configure SNMP protocol parameters which are generic to the SNMP instance.

### Syntax

**set instance** <instance-name> **protocol snmp** <attribute> <value>

| Attribute | Description |
|---|---|
| `version` | Specify the SNMP version. RBFS supports SNMP version 2c and version 3. You must first configure the desired version before configuring other functionalities. |
| `engine-id` | Specify the unique SNMP engine identifier. This is optional. If not specified, the system retrieves the default engine ID from the management port MAC address.<br><br>Every SNMP v3 agent includes an engine ID that is a unique identifier for the agent. The engine ID is used to provide a higher level of security using authentication and encryption for SNMP v3 messages. |

Example: SNMP Version and Engine Identifier Configuration

The following commands configure SNMP version 3 and engine ID: 268956.

```
set instance default protocol snmp version 3
set instance default protocol snmp engine-id 268956
```

The following example shows the SNMP version and engine ID configurations.

```
supervisor@rtbrick.net: cfg> show config instance default protocol snmp
{
  "rtbrick-config:snmp": {
    "version": "3",
    "engine-id": "268956"
  }
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### SNMP Community Configuration

An SNMP community can be defined only in the SNMP version 2c.

### Syntax

**set instance** <instance-name> **protocol snmp community**

| Attribute | Description |
|---|---|
| `access-mode` | Specify the access mode. Read, write and append are modes of access. 'ReadOnly' is the currently supported access mode. |
| `view` | Specify the list of view identifiers. View is optional. For information about Views, see section "2.2.3 SNMP View Configuration". |

Example: SNMP v2c Community Configuration

The following commands configure a Community named 'public' with read-only access right to the 'interfaces' View.

```
set instance default protocol snmp version 2c
set instance default protocol snmp community public access-mode ReadOnly
```

The following example shows SNMP v2c community configurations.

```
supervisor@rtbrick: cfg> show config instance default protocol snmp
{
  "rtbrick-config:snmp": {
    "version": "2c",
    "community": [
      {
        "name": "public"
        "access-mode": "ReadOnly"
      }
    ]
  }
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### SNMP View Configuration

An SNMP View is a subset of MIB objects. Views allow you to restrict access to certain items in the SNMP PDUs. You can restrict user and community access to certain attributes by defining views. A view restricts access to the PDUs included in the View. If the access is not restricted by views, the user or community is allowed to view all data available through SNMP.

**Syntax**

**set instance** <instance-name> **protocol snmp** <attribute> <value>

| Attribute | Description |
|---|---|
| `include <include>` | List of OID patterns that are included in the view. |
| `instance` | List of instances. It restricts the view to the specified instances. If no instance is defined, the view can access to all instances. |

Example: SNMP View Configuration

The following commands configure SNMP View. In this example configuration, SNMP version has been specified as 2c and 'View' name is specified as interfaces. The 'interfaces' view includes the OID 1.3.6.1.2.1.2.* in the view list. In addition, the configuration shows a user 'community' named public has been configured and the community has read-only access to the View.

```
set instance default protocol snmp version 2c
set instance default protocol snmp view interfaces include 1.3.6.1.2.1.2.*
set instance default protocol snmp community public access-mode ReadOnly
set instance default protocol snmp community public view interfaces
```

The following example shows the SNMP View configuration.

```
supervisor@rtbrick: cfg> show config instance default protocol snmp
{
  "rtbrick-config:snmp": {
    "version": "2c",
    "view": [
      {
        "name": "interfaces",
        "include": [
          "1.3.6.1.2.1.2.*"
          ]
      }
    ],
    "community": [
      {
        "name": "public",
        "access-mode": "ReadOnly",
        "view": [
          "interfaces"
          ]
      }
    ]
  }
}
```

To access the RESTCONF API that corresponds to this CLI, click here.

#### SNMP Trap Configuration

The trap is a notification about a specific condition or event that occurs on the device. RBFS supports event notifications for events: link up and link down. Unlike other PDU types, a trap is a notification that is sent without any request from the SNMP manager.

### Syntax

**set instance** <instance-name> **protocol snmp trap** <trap-name> <attribute> <value>

| Attribute | Description |
|---|---|
| `community` | Community is supported in SNMPv2c only. Specify the 'Community' to enable trap notification for SNMPv2c. |
| `include` | List of OIDs that are included for the trap notifications. |
| `instance` | Specify the instance. |
| `trap-receiver` | Specify trap receiver device. |
| `user-profile` | User profile is supported on SNMPv3 only. Specify the 'user profile' to enable trap notification for SNMPv3. |

To access the RESTCONF API that corresponds to this CLI, click here.

#### SNMP User Profile Configuration

You can create user profiles for SNMP version 3. It allows you to define login credentials, authentication methods, and privacy control.

### Syntax

**set instance** <instance-name> **protocol snmp user-profile**

| Attribute | Description |
|---|---|
| `authentication-protocol` | Specify SNMP authentication protocol. MD5, NoAuth, SHA, SHA224, SHA256, SHA384, and SHA512 are the supported authentication protocol. |
| `password-encrypted-text` | Specify SNMP user password in encrypted text. |
| `password-plain-text` | Specify SNMP user password in plain text. |
| `privacy-password-encrypted-text` | Specify SNMP privacy password in encrypted text. |
| `privacy-password-plain-text` | Specify SNMP privacy password in plain text. |

| Attribute | Description |
|---|---|
| `privacy-protocol` | Specify SNMP privacy protocol. Supported privacy protocols include AES192, AES192C, AES256, AES256C, DES, and NoPriv. |
| `security-level` | Specify SNMP v3 security level. Security levels exist only in SNMP v3. The following security levels are supported:<br>• noAuthNoPriv: no authentication, no privacy<br>• authNoPriv: authentication, no privacy<br>• authPriv: authentication, privacy |
| `view` | Specify SNMP view list. |

Example: SNMP User Profile Configuration

The following commands configure SNMP user profile. At first, SNMP Version 3 is configured with the user profile name as operator. Password type has been selected as password encrypted text. In this configuration, the security level is configured as `AuthNoPriv` and MD5 as type of the authentication protocol.

```
set instance default protocol snmp version 3
set instance default protocol snmp user-profile operator
set instance default protocol snmp user-profile operator password-encrypted-text
$2a6fd7db50a18a9f1f16b5c5b4214fab0
set instance default protocol snmp user-profile operator security-level AuthNoPriv
set instance default protocol snmp user-profile operator authentication-protocol MD5
```

The following example shows the SNMP User Profile Configuration

```
supervisor@rtbrick: cfg> show config instance default protocol snmp
{
  "rtbrick-config:snmp": {
    "version": "3",
    "user-profile": [
      {
        "name": "operator",
        "password-encrypted-text": "$2a6fd7db50a18a9f1f16b5c5b4214fab0",
        "security-level": "AuthNoPriv",
        "authentication-protocol": "MD5"
      }
    ]
  }
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### SNMP Location Configuration

With the Location option, you can configure basic administrative information such as the physical location for a managed device.

### Syntax

**set instance** <instance-name> **protocol snmp location** <location-name>

Example Command:

```
set instance default protocol snmp location BLR
```

#### SNMP Contact Configuration

With the SNMP Contact option, you can configure basic administrative information for a managed device. You can specify the contact information of the person responsible for the device.

### Syntax

**set instance** <instance-name> **protocol snmp contact** <contact-name>

Example Command:

```
set instance default protocol snmp contact RtBrick
```

### Examples for SNMP Walk Operation

#### SNMP v2c SNMP Walk Output

The following is a sample output for the SNMP Walk for SNMP version 2c. SNMP version 2c has been configured with Community name as 'public' and, host IP address as 10.200.134.25.

```
snmpwalk -v 2c -c public 10.200.134.25
iso.3.6.1.2.1.2.2.1.1.1 = INTEGER: 1
iso.3.6.1.2.1.2.2.1.1.2 = INTEGER: 2
iso.3.6.1.2.1.2.2.1.1.3 = INTEGER: 3
iso.3.6.1.2.1.2.2.1.1.4 = INTEGER: 4
iso.3.6.1.2.1.2.2.1.1.5 = INTEGER: 5
iso.3.6.1.2.1.2.2.1.1.6 = INTEGER: 6
iso.3.6.1.2.1.2.2.1.1.7 = INTEGER: 7
iso.3.6.1.2.1.2.2.1.1.8 = INTEGER: 8
iso.3.6.1.2.1.2.2.1.1.9 = INTEGER: 9
iso.3.6.1.2.1.2.2.1.1.10 = INTEGER: 10
iso.3.6.1.2.1.2.2.1.1.11 = INTEGER: 11
iso.3.6.1.2.1.2.2.1.1.12 = INTEGER: 12
```

```
iso.3.6.1.2.1.2.2.1.1.13 = INTEGER: 13
iso.3.6.1.2.1.2.2.1.1.14 = INTEGER: 14
iso.3.6.1.2.1.2.2.1.1.15 = INTEGER: 15
iso.3.6.1.2.1.2.2.1.1.16 = INTEGER: 16
iso.3.6.1.2.1.2.2.1.1.17 = INTEGER: 17
iso.3.6.1.2.1.2.2.1.1.18 = INTEGER: 18
iso.3.6.1.2.1.2.2.1.1.19 = INTEGER: 19
iso.3.6.1.2.1.2.2.1.1.20 = INTEGER: 20
iso.3.6.1.2.1.2.2.1.1.21 = INTEGER: 21
iso.3.6.1.2.1.2.2.1.1.22 = INTEGER: 22
iso.3.6.1.2.1.2.2.1.1.23 = INTEGER: 23
<...>
```

**SNMP v3 SNMP Walk Output**

The following is a sample output for the SNMP Walk for SNMP version 3. SNMP version 3 has been configured with user as 'operator', MD5 as the authentication protocol, authNoPriv as the security level, and 10.200.134.25 as the host IP address.

```
snmpwalk -v 3 -u operator -A operator -a MD5 -l authNoPriv 10.200.134.25
iso.3.6.1.2.1.2.2.1.1.1 = INTEGER: 1
iso.3.6.1.2.1.2.2.1.1.2 = INTEGER: 2
iso.3.6.1.2.1.2.2.1.1.3 = INTEGER: 3
iso.3.6.1.2.1.2.2.1.1.4 = INTEGER: 4
iso.3.6.1.2.1.2.2.1.1.5 = INTEGER: 5
iso.3.6.1.2.1.2.2.1.1.6 = INTEGER: 6
iso.3.6.1.2.1.2.2.1.1.7 = INTEGER: 7
iso.3.6.1.2.1.2.2.1.1.8 = INTEGER: 8
iso.3.6.1.2.1.2.2.1.1.9 = INTEGER: 9
iso.3.6.1.2.1.2.2.1.1.10 = INTEGER: 10
iso.3.6.1.2.1.2.2.1.1.11 = INTEGER: 11
iso.3.6.1.2.1.2.2.1.1.12 = INTEGER: 12
iso.3.6.1.2.1.2.2.1.1.13 = INTEGER: 13
iso.3.6.1.2.1.2.2.1.1.14 = INTEGER: 14
iso.3.6.1.2.1.2.2.1.1.15 = INTEGER: 15
iso.3.6.1.2.1.2.2.1.1.16 = INTEGER: 16
iso.3.6.1.2.1.2.2.1.1.17 = INTEGER: 17
iso.3.6.1.2.1.2.2.1.1.18 = INTEGER: 18
iso.3.6.1.2.1.2.2.1.1.19 = INTEGER: 19
iso.3.6.1.2.1.2.2.1.1.20 = INTEGER: 20
iso.3.6.1.2.1.2.2.1.1.21 = INTEGER: 21
iso.3.6.1.2.1.2.2.1.1.22 = INTEGER: 22
iso.3.6.1.2.1.2.2.1.1.23 = INTEGER: 23
<...>
```

The following is a sample output for the SNMP System MIB walk operation.

```
snmpwalk -v 2c -c ufi17.q2c.u34.r6.nbg.rtbrick.net"
<...>
```

```
iso.3.6.1.2.1.1.1 = STRING: "UfiSpace S9600-102XC-R, RBFS 24.9.0-
g6daily.20241007154425+Bdevelopment.C8e6a3a56"
iso.3.6.1.2.1.1.2 = OID: iso.3.6.1.4.1.50058.1.1
iso.3.6.1.2.1.1.3 = Timeticks: (35079) 0:05:50.79
iso.3.6.1.2.1.1.4 = ""
iso.3.6.1.2.1.1.5 = STRING: "ufi17.q2c.u34.r6.nbg.rtbrick.net"
iso.3.6.1.2.1.1.6 = ""
iso.3.6.1.2.1.1.7 = INTEGER: 12
iso.3.6.1.2.1.1.7 = No more variables left in this MIB View (It is past the end
of the MIB tree)
<...>
```

Example 2:

```
snmpwalk -v 2c -c rbfs ufi11.q2a.u15.r6.nbg.rtbrick.net
iso.3.6.1.2.1.1.1 = STRING: "Ufi Space S9510-28DC-B, RBFS 25.1.0-
g6daily.20250415154806+Bdevelopment.Cdf3d9037"
iso.3.6.1.2.1.1.2 = OID: iso.3.6.1.4.1.50058.1.1
iso.3.6.1.2.1.1.3 = Timeticks: (41382) 0:06:53.82
iso.3.6.1.2.1.1.4 = STRING: "RtBrick"
iso.3.6.1.2.1.1.5 = STRING: "ufi11.q2a.u15.r6.nbg.rtbrick.net"
iso.3.6.1.2.1.1.6 = STRING: "BLR"
iso.3.6.1.2.1.1.7 = INTEGER: 12
iso.3.6.1.2.1.1.7 = No more variables left in this MIB View (It is past the end of the
MIB tree)
```

# 11. Infrastructure

Infrastructure modules provide foundational capabilities for managing and maintaining carrier-grade network environments. Together, the infrastructure features form the backbone of a robust network.

Key Features:

- NTP: Provides precise time synchronization across all devices.
- Clock Synchronization: Centralized management of system clocks to ensure consistency for protocols and services that depend on precise timing on access-leaf images.
- Logging: Comprehensive logging framework for capturing system, security, and operational events for troubleshooting.
- LED Control: Enables management of chassis and interface LEDs to quickly identify hardware status and perform diagnostics.
- IPMI : Facilitates out-of-band hardware monitoring, management, and recovery to ensure hardware health and remote troubleshooting.
- Inband Management: Provides secure, efficient management of network devices.
- DHCP Relay/Proxy: Provides flexible DHCP relay and proxy functionality to support dynamic IP address assignment.

The following guides are available under the Infrastructure section:

- NTP
- Clock Synchronization
- Logging
- LED
- IPMI
- In-band Management
- DHCP Relay Overview

## 11.1. NTP

### 11.1.1. NTP Overview

Network Time Protocol (NTP) provides accurate time across a whole network of devices such as routers or switches and synchronizes time among these devices on a network. NTP uses an NTP server that maintains a highly accurate time. An NTP network is comprised of devices (clients) that are to be synchronized with the NTP server that has UTC time and provides it to the client devices.

**Prerequisites**

- Install and configure an NTP server on your network and remember the IP address of the NTP

server.

#### Supported Platforms

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

### 11.1.2. NTP Configuration

#### Configuration Syntax and Commands

The following sections describe the NTP configuration syntax and commands.

##### Specifying the NTP Server

To configure the NTP server, you must specify details such as the IPv4 address and the domain name of the NTP server.

#### Syntax

**set system ntp server** <server-id> <option>

| Attribute | Description |
|---|---|
| `server <server-id>` | Specifies the name of the NTP server. |
| `ipv4-address <ipv4-address>` | Specifies the IPv4 address of the NTP server. |
| `domain-name <domain-name>` | Specifies the domain name of the NTP server. |
| `minpoll` | Specifies the minimum interval between requests in seconds. NTP dynamically selects the optimal poll interval between the values specified for `minpoll` and `maxpoll`. |
| `maxpoll` | Specifies the maximum interval between requests in seconds. NTP dynamically selects the optimal poll interval between the values specified for `minpoll` and `maxpoll`. |

Example:

```
{
  "ietf-restconf:data": {
    "rtbrick-config:system": {
      "ntp": {
        "server": [
          {
            "server-name": "NtpServerIPv4",
            "ipv4-address": "192.168.16.1",
            "minpoll": 3,
            "maxpoll": 17
          }
```

```
        ]
      }
    }
  }
}
```

#### NTP Service Configuration

You can enable the NTP service under a specific VRF where NTP server is reachable (usually, the inband management VRF).

**Syntax:**

**set inband-management instance** <name> **ntp**

| Attribute | Description |
|---|---|
| `<name> <instance-name>` | Name of the instance |
| `true/false` | Enable or disable Network Time Protocol. |

Example: NTP Configuration

```
{
    "rtbrick-config:inband-management": {
      "instance": [
        {
          "name": "mgmt-vrf",
          "ntp": "true"
        }
      ]
    }
  }
```

#### Setting the System Date & Time Using DHCP Option 42 NTP Servers Information

The RtBrick RtBrick Host installer image performs a boot time ntpdate operation to set the system date and time.

By default, it uses the pool.ntp.org but DHCP bindings received over the ma1 interface will override it if DHCP option 42 contains a list of one or more NTP servers.

Separately from this boot time setting, the system will continue to use NTPd to maintain accurate system date and time using the list of NTP servers contained in the RBFS configuration.

The `/etc/ntp.boot-time-servers.conf` file can be used to verify as its contents will be updated with the NTP servers from DHCP. Also, logs are available in `/var/log/rtbrick-boot-time-ntpdate.log`.

#### Enabling RBFS Logs to Monitor NTP Peer Events

You can enable RBFS logs to monitor NTP peer events, allowing RBFS to fetch peer statistics using NTP Mode 6 control messages. Additionally, the "show" commands have been improved to offer information on NTP status, associations, history, and more detailed insights.

The configuration below enables RBFS logs for NTP peer events. It sets the log module to "resmon," specifies the group as "resmon-ntp-peer-events," sets the log level to "info," and sets the rate limit to "0."

```
set log module resmon
set log module resmon group resmon-ntp-peer-events
set log module resmon group resmon-ntp-peer-events level info
set log module resmon group resmon-ntp-peer-events rate-limit 0
```

### 11.1.3. NTP Operational Commands

#### OSPF Show Commands

##### NTP Status

Displays the NTP status information.

**Syntax:**

**show ntp status**

```
supervisor@rtbrick: op> show ntp status
associd=0 status=C016 leap_alarm, sync_unspec, 1 event, restart,
leap=3, stratum=16, precision=-21, rootdelay=0.000, rootdisp=0.225,
refid=INIT, reftime=0x00000000.00000000  1900-01-01T00:00:00+00:00,
tc=0, peer=0, offset=0.000000,
frequency=0.000000, sys_jitter=0.000000, clk_jitter=0.000477,
clock=0xedcd4762.b7eb15fc  2026-06-05T13:02:26.718430+00:00, processor="x86_64", system="Linux/6.1.170",
version="ntpd ntpsec-1.2.2", clk_wander=0.000000,
leapsec=2017-01-01T00:00Z, expire=2026-12-28T00:00Z, mintc=0
```

##### NTP Peer

Displays history of the NTP peers, showing when they went up or down and any changes in their status, identified by the peer's IP address.

**Syntax:**

**show ntp peer** <options>

| Option | Description |
|---|---|
| `<peer-ip>` | IP address of the NTP peer to show history for. |
| `history` | Displays history of the NTP peers, showing when they went up or down and any changes in their status, identified by the peer's IP address. |
| `detail` | Displays detailed information for a specific NTP peer, including its association ID, status, source address, port, hostname, destination address, leap indicator, stratum, precision, root delay, root dispersion, reference ID, reference time, receive time, reachability, host mode, peer mode, poll intervals, headway, key ID, offset, delay, dispersion, and jitter. |
| `<peer-ip> history` | Displays the history of a specific NTP peer. |
| `<peer-ip> detail` | Displays detailed information for a specific NTP peer. |

The example below shows the history of a specific NTP peer, showing when it went up or down and any changes in its status.

```
supervisor@rtbrick: op> show ntp peer 192.0.2.105 history
srcadr                          timestamp                                 associd     status    reason
192.0.2.105                     Fri Jun 05 13:01:36 GMT +0000 2026          17767       8011    Peer Up
192.0.2.105                     Fri Jun 05 13:01:56 GMT +0000 2026          17767       8011    Peer Down
192.0.2.105                     Fri Jun 05 13:02:11 GMT +0000 2026          17767       8011    Peer Up
```

The example below shows detailed information of the NTP peers using NTP Mode 6 control message READVAR, including their association ID, status, source address, port, hostname, destination address, leap indicator, stratum, precision, root delay, root dispersion, reference ID, reference time, receive time, reachability, host mode, peer mode, poll intervals, headway, key ID, offset, delay, dispersion, and jitter.

```
supervisor@rtbrick>LEAF01: cfg> show ntp peer detail
associd: 60555, status: 161a
  srcadr: 198.51.100.13, srcport: 123
  srchost: node-2.infogral.is
  dstadr: 198.51.100.101, Port: 123
  leap: 0 leap_none, stratum: 2, precision: -25
  rootdelay: 5.112
  rootdisp: 0.290
  refid: 192.0.2.4
  reftime: 2025-05-27T12:34:07.000003+0000
  rec: 2025-05-27T12:36:32.000003+0000
  reach: 336, unreach: 2
  hmode: 3 pmode: 4, hpoll: 9, ppoll: 9
  headway: 0, keyid: 0
  flash: 0 Ok
  offset: -0.198
  delay: 2.828
```

```
  dispersion: 16.124001
  jitter: 0.638
  xleave: 0.036000
<...>
```

#### NTP Associations

Displays the information about the various NTP servers configured.

**Syntax:**

**show ntp association** <options>

| Option | Description |
|---|---|
| `no-resolve` | Details without DNS resolution. |

```
show ntp association
```

```
supervisor@rtbrick: op> show ntp association
Flags: space ( ) Discarded due to high stratum and/or failed sanity checks, x
Designated falseticker by the intersection algorithm,
       . Culled from the end of the candidate list, - Discarded by the clustering
algorithm,
       + Included in the final selection set, # Selected for synchronization but
distance exceeds maximum,
       * Selected for synchronization, o Selected for synchronization, pps signal in
use.
remote                           refid            st t  when  poll reach    delay
offset  jitter
 192.0.2.105                     .INIT.           16 u     -    64     0    0.000
0.000   0.000
 192.0.2.106                     .INIT.           16 u     -    64     0    0.000
0.000   0.000
```

#### System Uptime

The `show system uptime` command provides a timeline summary of the system, including when it booted, when forwarding started, last configuration change, and current CPU load.

**Syntax:**

**show system uptime**

Example for the command:

```
supervisor@rtbrick.net: cfg> show system uptime
Time source: NTP CLOCK, Current time: 2026-04-02 04:46:30 UTC
Forwarding started: 2026-04-01 11:48:23 UTC (16 hours, 58 minutes ago)
Last configured: 2026-04-02 04:45:08 UTC (1 minute ago) by supervisor
Container boot time: 2026-04-01 11:08:55 UTC (17 hours, 37 minutes ago)
CPU load averages [1m, 5m, 15m]: 1.58, 1.56, 3.32
```

**Verifying NTP Service on Linux**

To get more information about NTP's status, use the `ntpq` command:

The command displays the NTP servers that the system is synchronized with.

```
supervisor@rtbrick>LEAF01:~ $ sudo ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
198.51.100.5       .XFAC.          16 u    -   64    0    0.000    0.000   0.000
```

# 11.2. Clock Synchronization

## 11.2.1. Clock Synchronization

**Overview**

**Overview**

This guide provides an overview of high-precision clock synchronization mechanisms, including the Precision Time Protocol (PTP), a protocol for clock synchronization supported in RBFS, and covers its configuration and operational commands. PTP is used to synchronize clocks across a network with high precision, achieving accuracy in the nanosecond range.

PTP synchronizes clocks by exchanging timestamped messages between devices. The slave device calculates the clock offset and network delay, then continuously adjusts its clock using a servo algorithm for smooth and precise synchronization. A Hardware Clock (PHC) ensures nanosecond precision for timestamps.

The diagram below illustrates how the **master clock** serves as the reference time source, and the **slave clocks** (Switch-1 and Switch-2) synchronize their time to that of the master.

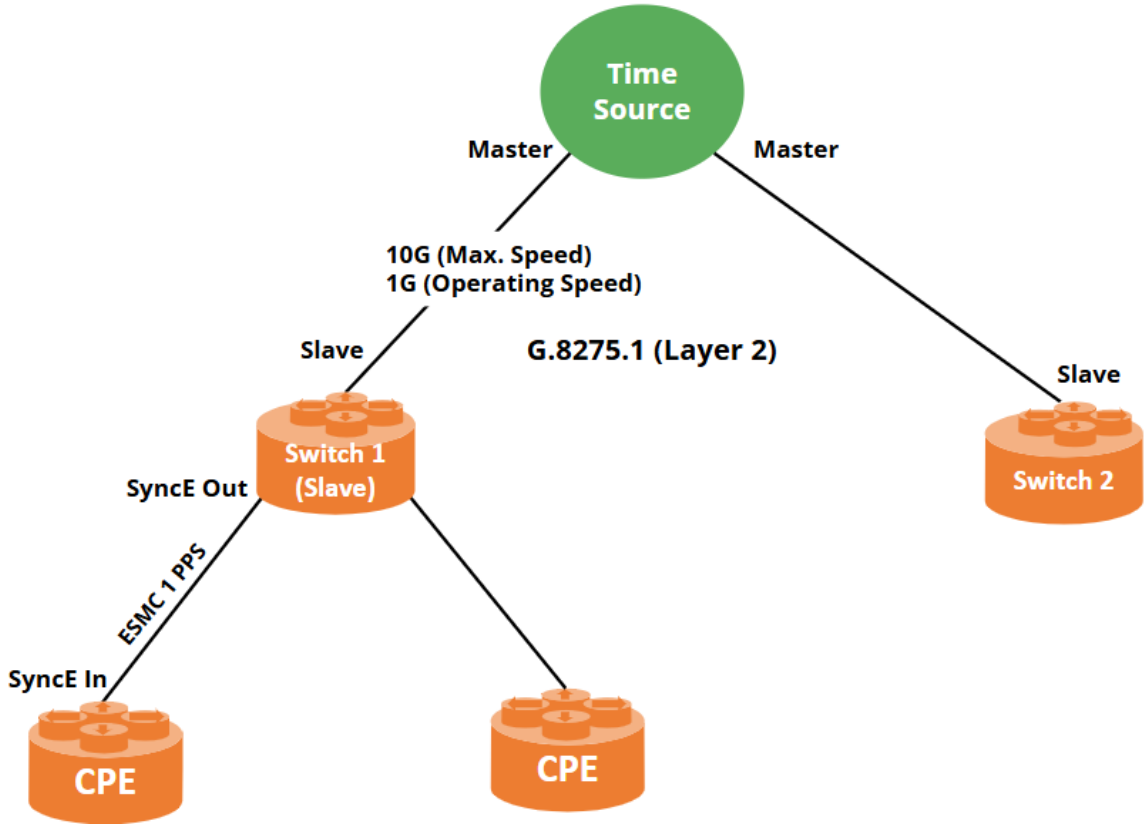



*Figure 162. Clock Synchronization Overview*

RBFS supports the following PTP profiles and standards:

- IEEE 1588-2008: Describes the IEEE 1588-2008 standard for Precision Time Protocol (PTP) used in clock synchronization.
- ITU-T G.8275.1: Defines the timing requirements for packet-based networks to support time-sensitive applications.

In addition, RBFS supports PTP operation at Layer 2.

### SyncE ESMC Support

SyncE (Synchronous Ethernet) and ESMC (Ethernet Synchronization Messaging Channel) are crucial components for precise time synchronization in networks.

SyncE provides a mechanism for distributing frequency synchronization over Ethernet at the physical layer (Layer 1). It ensures that all connected network elements operate on the same clock frequency, preventing clock drifts and maintaining a stable timing reference.

ESMC is a Layer 2 protocol (EtherType: 0x8809) that complements SyncE by carrying Synchronization Status Messages (SSMs). These messages convey information about the quality of the clock source being used by SyncE. ESMC allows network devices to select the best available clock source and to automatically switch to a different, higher-quality source if the current one degrades or fails.

### Licensing Requirements

Clock synchronization requires a separate license to be procured from RtBrick, in addition to other

licenses.

#### Supported Platforms

Not all features are supported on every hardware platform. Refer to the Feature Support Matrix section of the *Platform Guide* for details about which features and sub-features are supported on each platform.

### 11.2.2. Clock Synchronization Configuration

Clock Synchronization configuration involves global and interface-level settings.

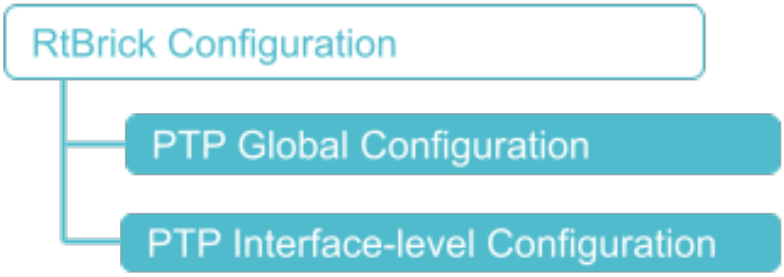



*Figure 163. Clock Synchronization Configuration Hierarchy*

#### Global Configuration

These commands configure the global clock Synchronization settings for the system.

**Syntax**:

```
set system ptp clock <options>
```

**Attribute Description:**

| Attribute | Description |
|---|---|
| `clock g8275.1` | Enables ITU-T G.8275.1 clock profile for PTP.<br>As per ITU-T G.8275.1 profile, the default domain value for the PTP is 24. |
| `clock g8275.1 mode (slave \| boundary)` | Configures the PTP clock to operate in slave or boundary mode.<br>**slave**: The device synchronizes its local clock to an upstream master.<br>**boundary**:The device acts as both a slave (upstream) and a master (downstream), relaying and conditioning timing information. |

| Attribute | Description |
| --- | --- |
| `two-step enable` | Enables two-step PTP synchronization in which the device's PTP clock to use two-step synchronization rather than one-step synchronization. In two-step synchronization mode, the device first sends a Sync message, captures the exact hardware transmission timestamp, and then immediately sends that timestamp in a separate Follow_Up message. |
| `one-pps-out tod enable` | Enables one-PPS output with TOD functionality. When enabled, the device outputs a 1PPS signal with ToD information via the RJ45 port, providing a means to verify that the clock is correctly frequency-locked and phase-aligned. |

The example below shows the configuration for enabling PTP (Precision Time Protocol) in `slave` mode using the `g8275.1` profile.

```
set system ptp clock g8275.1 mode slave
```

```
supervisor@rtbrick.net: cfg> show config system ptp
{
  "rtbrick-config:ptp": {
    "clock": [
      {
        "profile": "g8275.1",
        "mode": "slave"
      }
    ]
  }
}
```

The example below shows the configuration for enabling PTP (Precision Time Protocol) in boundary mode using the g8275.1 profile.

```
set system ptp clock g8275.1 mode boundary
```

```
supervisor@rtbrick.net: cfg> show config system ptp
{
  "rtbrick-config:ptp": {
    "clock": [
      {
        "profile": "g8275.1",
        "mode": "boundary"
      }
    ]
  }
```

```
}
```

#### Interface-level Configuration

These commands enable and configure PTP on specific network interfaces.

**Syntax**:

**set interface** <interface-name> <attribute> <value>

| Attribute | Description |
|---|---|
| `<interface-name>` | Name of the network interface. |
| `ptp enable true` | Enables (true) clock synchronization for the specified interface |
| `synce enable true` | Enables (true) synchronous Ethernet for the specified interface |
| `synce esmc source output` | Defines the source for the ESMC messages to be output by the interface. |
| `synce esmc transmit-quality` | Sets the base configuration for how the interface communicates quality levels (QL) via ESMC to downstream devices. |
| `synce esmc transmit-quality dnu-force` | Configures the interface to advertise a "Do Not Use" (DNU) quality level, effectively preventing downstream devices from using this interface as a timing source. |

After the clock module is enabled and the clock is configured on the device, you can enable `ptp`, `synce`, and `synce esmc` on the network interface.

The following configuration enables PTP on the network interface named "ifp-0/1/66".

```
set interface ifp-0/1/66
set interface ifp-0/1/66 ptp enable true
```

```
supervisor@rtbrick.net: cfg> show config interface ifp-0/1/66
{
  "rtbrick-config:interface": [
    {
      "name": "ifp-0/1/66",
      "ptp": {
        "enable": "true"
      }
    }
  ]
}
```

The following configuration enables `synce` on the network interface named "ifp-0/1/66".

```
set interface ifp-0/1/66
set interface ifp-0/1/66 sync enable true
```

```
supervisor@rtbrick.net: cfg> show config interface ifp-0/1/66
{
  "rtbrick-config:interface": [
    {
      "name": "ifp-0/1/66",
      "sync": {
        "enable": "true"
      }
    }
  ]
}
```

The following configuration enables `synce esmc` on the network interface named "ifp-0/1/66".

```
set interface ifp-0/1/66
set interface ifp-0/1/66 sync synce esmc true
```

```
supervisor@rtbrick.net: cfg> show config interface ifp-0/1/66
{
  "rtbrick-config:interface": [
    {
      "name": "ifp-0/1/66",
      "sync": {
        "esmc": "true"
      }
    }
  ]
}
```

#### PTP and SyncE ESMC Destination MAC Configuration

You can configure the destination MAC for outgoing SyncE ESMC and PTP packets. The following CLI commands allow you to configure the destination MAC address for SyncE ESMC packets.

**Syntax:**

**set system synce esmc destination-mac-address** <destination-mac-address>

**set system ptp global destination-mac-address** <destination-mac-address>

| Attribute | Description |
| --- | --- |
| `<destination-mac-address>` | Specifies the destination MAC address for outgoing SyncE and PTP packets.<br>If no destination MAC address is configured, the system will use the default MAC address. |

The following example sets the destination MAC address for SyncE ESMC packets to 01:80:C2:00:00:02.

```
set system synce esmc destination-mac-address 01:80:C2:00:00:02
```

```
{
  "rtbrick-config:system": {
    "synce": {
      "esmc": {
        "destination-mac-address": "01:80:C2:00:00:02"
      }
    }
  }
}
```

The following example sets the destination MAC address for PTP packets to 01:1B:19:00:00:00.

```
set system ptp global destination-mac-address 01:1B:19:00:00:00
```

```
{
  "rtbrick-config:system": {
    "ptp": {
      "global": {
        "destination-mac-address": "01:1B:19:00:00:00"
      }
    }
  }
}
```

#### Configuring SyncE Input Source for PTP Interfaces

You can configure whether a specific PTP interface serves as the physical frequency input source for SyncE.

**Syntax:**

**set system sync interface-ptp input-source ( disable | enable )**

| Attribute | Description |
|---|---|
| `input-source (disable \| enable)` | Specifies whether the PTP interface is used as the physical SyncE frequency input source. Set to **enable** to use the interface as the SyncE input source, or **disable** to stop using it as the SyncE input source. By default, this is enabled. |

## 11.2.3. Clock Synchronization Operational Commands

### Validating the Configurations Using the ufi-ptp-cli Shell

To validate the configuration, use the ufi-ptp-cli shell as shown in the following example.

The following example displays the output of the show ptp clock info command, detailing the PTP clock's configuration. It includes information such as the clock identity, profile type (G.8275.1), clock type (Time Slave Clock), and various PTP interval settings.

```
supervisor@rtbrick>rtbrick.net:~ $ ufi-ptp-cli
Check UFI PTP Service Status ... Ready.
Read configuration successfully
Build On Jun 3 2025 02:14:45
ufi-ptp> enable
ufi-ptp# show ptp clock info
Clock Identity                   : 5C:07:58:A2:CE:5C:00:01
Profile                          : G.8275.1 profile
Clock Type                       : Time Slave Clock
Servo Alg                        : Full-On-Path
Number Of Ports                  : 32
Clock Class                      : 255
Domain Number                    : 24
Scaled Log Variance              : 0xFFFF
Priority1                        : 128
Priority2                        : 255
Slave Only                       : True
Two Step                         : False
Announce Receipt Timeout (min)   : 3
Announce Receipt Timeout (def)   : 3
Announce Receipt Timeout (max)   : 10
Log Announce Interval (min)      : -3
Log Announce Interval (def)      : -3
Log Announce Interval (max)      : -3
Log Sync Interval (min)          : -4
Log Sync Interval (def)          : -4
Log Sync Interval (max)          : -4
Log Min Delay Req Interval (min) : -4
Log Min Delay Req Interval (def) : -4
Log Min Delay Req Interval (max) : -4
```

```
Domain # (min)                  : 24
Domain # (def)                  : 24
Domain # (max)                  : 43
Priority 1                      : 128
Priority 2 (min)                : 0
Priority 2 (def)                : 128
Priority 2 (max)                : 225
Local Priority                  : 128
```

The following example displays the output of the `show ptp port 67 peer` command, detailing PTP packet statistics for peer `0` on `Port 67`. It provides counts of various received and transmitted PTP messages, such as Sync, Delay Request, Delay Response, and Announce messages.

```
ufi-ptp# show ptp port 9 peer
Port 9:
  Peer #0
  L2 address                      : D8:3A:DD:AA:4D:9D
  Clock Identity                  : D8:3A:00:FF:FE:AA:4D:9D
  Received Sync                   : 138913
  Received Delay Request          : 0
  Received Delay Response         : 138904
  Received Peer Delay Request     : 0
  Received Peer Delay Response    : 0
  Received Followup               : 138913
  Received Peer Delay Followup    : 0
  Received Announce               : 69457
  Received Signalling             : 0
  Received Management             : 0
  Transmitted Sync                : 0
  Transmitted Delay Request       : 138904
  Transmitted Delay Response      : 0
  Transmitted Peer Delay Request  : 0
  Transmitted Peer Delay Response : 0
  Transmitted Followup            : 0
  Transmitted Peer Delay Followup : 0
  Transmitted Announce            : 0
  Transmitted Signalling          : 0
  Transmitted Management          : 0
```

The `show ptp port 67 master` command output displays that port 67 has one PTP master identified as E1:C1:1A:G5:L2:15.

```
ufi-ptp# show ptp port 67 master
Port 67:
Master #0 : E1:C1:1A:G5:L2:15
ufi-ptp#
```

### PTP Show CLIs

#### show ptp clock-info

This command displays detailed information about the PTP clock running on the device.

**Syntax:**

```
show ptp clock-info
```

The following example displays detailed information about the PTP clock. Key details include the clock identifier, profile (g8275.1), clock type (Slave Clock), domain number, and various PTP interval settings.

```
supervisor@rtbrick>switch: op> show ptp clock-info
Clock Information
  Clock identifier: 5c:07:58:a2:ce:5c:00:01
  Profile: g8275.1
  Clock type: Slave Clock
  Max number of ports: 32
  Clock class: 255
  Domain number: 24
  Scaled log variance: 0xFFFF
  Priority1: 128
  Priority2: 255
  Slave only: True
  Announce receipt timeout (Min): 3
  Announce receipt timeout (Def): 3
  Announce receipt timeout (Max): 10
  Log announce interval (Min): -3
  Log announce interval (Def): -3
  Log announce interval (Max): -3
  Log sync interval (Min): -4
  Log sync interval (Def): -4
  Log sync interval (Max): -4
  Log delay req interval (Min): -4
  Log delay req interval (Def): -4
  Log delay req interval (Max): -4
  Domain min: 24
  Domain default: 24
  Domain max: 43
  Priority2 min: 0
  Priority2 default: 128
  Priority2 max: 225
  Local priority: 128
  PTP Timescale: False
```

#### show ptp clock dataset

This command displays detailed information about the PTP clock dataset, including the current, default, parent, and time-properties datasets.

**Syntax**:

```
show ptp clock dataset current
show ptp clock dataset default
show ptp clock dataset parent
show ptp clock dataset time-properties
```

The following table provides the new command options and descriptions.

| Option | Description |
|---|---|
| `current` | This option provides a real-time view of the PTP clock dataset, specifically tracking dynamic metrics such as Mean Path Delay, Offset from Master, and Steps Removed. |
| `default` | Displays the default PTP clock dataset, including Clock Quality, Domain Number, and other largely static parameters. |
| `parent` | Displays parent clock information, including parent clock port, parent clock identity, and parent clock quality. Parent clock quality is sent to slaves when the device is acting as a PTP master on this port. |
| `time-properties` | Displays time properties, including current UTC offset, leap second status, and time traceable status. |

The following example output displays the current PTP clock information, including dynamic metrics such as Mean Path Delay, Offset from Master, and Steps Removed.

```
supervisor@rtbrick.net: op> show ptp clock dataset current
Offset from master: -294937568.000000 nsec
Mean path delay: 577.000000 nsec
Steps removed: 2
```

The following example output displays the default PTP clock information, including static parameters such as clock identity and clock quality.

```
supervisor@rtbrick.net: op> show ptp clock dataset default
Two step flag: No
Clock identity: 5c:07:58:a2:ce:5c:00:01
Number of ports: 32
Clock priority 1: 128
Clock priority 2: 255
Slave only: Yes
```

```
Local priority: 128
Max step removed: 255
Domain number: 24
Clock Quality
  Clock class: 255
  Clock accuracy: 0xFE
  Clock offset: 0xFFFF
```

The following example output displays the parent PTP clock information, including parent clock identity, parent clock class and parent clock accuracy.

```
supervisor@rtbrick.net: op> show ptp clock dataset parent
Parent Port
  Clock identity: d8:3a:00:ff:fe:aa:4d:9d
  Port Number: 2
Parent stats: No
Observed parent offset: Invalid
Observed parent phase change rate: Invalid
Grandmaster identity: d8:3a:00:ff:fe:aa:4d:9d
Grandmaster clock priority 1: 128
Grandmaster clock priority 2: 0
Grandmaster Clock Quality
  Clock class: 255
  Clock accuracy: 0x20
  Clock offset: 0xFFFF
```

The following example output displays the time properties dataset information, including current UTC offset, leap second status, and time traceable status.

```
supervisor@rtbrick.net: op> show ptp clock dataset time-properties
Current UTC offset: 0
Current UTC offset valid: True
Leap 59: False
Leap 61: False
Time traceable: True
Frequency traceable: True
PTP Timescale: False
Time source: GPS
Time of day: Wed Jun 03 2026 11:29:52.814493 UTC
```

#### show ptp servo

This command displays the status and configuration of the PTP servo mechanism on a device. The servo is a critical component in PTP that adjusts the local clock to synchronize with a master clock.

**Syntax:**

**show ptp servo**

The following example displays the output of the `show ptp servo` command, providing the PTP servo statistics for Clock 0. It details the servo's configuration, current state ("Frequency Locked"), lock status ("Locked"), and various correction and packet rate values.

```
supervisor@rtbrick>switch: op> show ptp servo
Servo Stat for Clock 0
  Servo config: Freq + Phase Correction
  Servo state: Frequency Locked
  Servo state duration: 2:49:37
  Lock status: Locked
  APTS mode: N/A
  Frequency correction: 12000.000000 ppb
  Phase correction: 542500000.000000 nsec
  Offset from master: -542543518.000000 nsec
  Mean path delay: 458.000000 nsec
  APTS GPS to PTP offset: N/A
  Sync packet rate: 16
  Delay packet rate: 16
```

#### show ptp servo history

The "show ptp servo history" command displays detailed PTP timing information and servo state changes. This command displays the most recent 100 state changes.

**Syntax:**

**show ptp servo history**

The following example output provides a log of PTP servo states, including specific lock statuses, correction metrics, and timing data recorded for a device.

```
supervisor@rtbrick.net: cfg> show ptp servo history
Total number of recorded changes: 6
Servo recorded at 2026-06-12 07:45:52
  Servo config: Freq + Phase Correction
  Servo state: Frequency Locked
  Servo state duration: 0:00:24
  Lock status: Locked
  APTS mode: N/A
  Frequency correction: 48.000000 ppb
  Phase correction: 294500000.000000 nsec
  Offset from master: -4000206155.000000 nsec
  Mean path delay: 299.000000 nsec
  APTS GPS to PTP offset: N/A
```

```
  Sync packet rate: 16
  Delay packet rate: 16
Servo recorded at 2026-06-12 07:45:22
  Servo config: Freq + Phase Correction
  Servo state: Holdover Out Of Spec
  Servo state duration: 0:00:04
  Lock status: Holdover
  APTS mode: N/A
  Frequency correction: 11125.000000 ppb
  Phase correction: -294750000.000000 nsec
  Offset from master: -4294720968.000000 nsec
  Mean path delay: 205.000000 nsec
  APTS GPS to PTP offset: N/A
  Sync packet rate: 16
  Delay packet rate: 16
Servo recorded at 2026-06-12 07:40:21
  Servo config: Freq + Phase Correction
  Servo state: Frequency Locked
  Servo state duration: 0:00:23
  Lock status: Locked
  APTS mode: N/A
  Frequency correction: 42.000000 ppb
  Phase correction: -294500000.000000 nsec
  Offset from master: -294883100.000000 nsec
  Mean path delay: 292.000000 nsec
  APTS GPS to PTP offset: N/A
  Sync packet rate: 16
  Delay packet rate: 16
Servo recorded at 2026-06-12 07:39:51
  Servo config: Freq + Phase Correction
  Servo state: Holdover Out Of Spec
  Servo state duration: 0:00:03
  Lock status: Holdover
  APTS mode: N/A
  Frequency correction: -11125.000000 ppb
  Phase correction: 294750000.000000 nsec
  Offset from master: -282462.000000 nsec
  Mean path delay: 391.000000 nsec
  APTS GPS to PTP offset: N/A
  Sync packet rate: 16
  Delay packet rate: 16
Servo recorded at 2026-06-12 07:05:49
  Servo config: Freq + Phase Correction
  Servo state: Frequency Locked
  Servo state duration: 0:00:21
  Lock status: Locked
  APTS mode: N/A
  Frequency correction: 12000.000000 ppb
  Phase correction: 294500000.000000 nsec
  Offset from master: -141093.000000 nsec
  Mean path delay: 204.000000 nsec
```

```
  APTS GPS to PTP offset: N/A
  Sync packet rate: 16
  Delay packet rate: 16
Servo recorded at 2026-06-12 07:04:49
  Servo config: Freq + Phase Correction
  Servo state: Acquiring
  Servo state duration: 0:00:23
  Lock status: Acquiring
  APTS mode: N/A
  Frequency correction: 5875.000000 ppb
  Phase correction: 294500000.000000 nsec
  Offset from master: -170559.000000 nsec
  Mean path delay: 238.000000 nsec
  APTS GPS to PTP offset: N/A
  Sync packet rate: 16
  Delay packet rate: 16
```

#### show ptp statistics

This command displays the PTP packet statistics for a device.

**Syntax:**

**show ptp statistics** <interface>

| Option | Description |
|---|---|
| - | Without any option, the commands displays the PTP packet statistics for all interfaces. |
| <interface> | Displays the PTP packet statistics for the specified interface. |

The following example displays the output of the show ptp statistics command, detailing both global and interface-specific (ifp-0/1/8) PTP packet transmission, reception, and discard counts.

```
supervisor@rtbrick>switch: op> show ptp statistics
Total PTP statistics
  Tx: 161895   Rx: 566652   Discarded: 16
Interface           Tx             Rx             Discarded
ifp-0/1/8           161895         566652         16
```

The following example displays PTP packet statistics for interface ifp-0/1/8, showing 161895 packets transmitted, 566652 received, and 16 discarded.

```
supervisor@rtbrick>switch: op> show ptp statistics interface ifp-0/1/8
ifp-0/1/8
  Tx: 161895
  Rx: 566652
```

```
  Discarded: 16
```

**show ptp interface <interface-name>**

The show ptp interface <interface-name>` command allows you to retrieve specific PTP information for a specified interface.

**Syntax:**

**show ptp interface** <interface-name> (**dataset** | **master** | **peer** | **slave**)

| Attribute | Description |
|---|---|
| `interface <interface-name> (dataset \| master \| peer \| slave)` | Specifies the interface for which the PTP information is displayed.<br>• **dataset**: Displays the operational data set for the specified PTP port, including its current configuration and operational status.<br>• **master**: Shows details regarding the PTP master clock associated with that interface.<br>• **peer**: Provides statistics and configuration details for the PTP peer connected to the interface.<br>• **slave**: Displays status and configuration information for the interface when it is operating in a slave role. |

The example output below displays the detailed operational dataset for the PTP interface ifp-0/1/8, providing details about its current port state, configuration parameters, and information about the associated foreign master clock.

```
supervisor@rtbrick.net: cfg> show ptp interface ifp-0/1/8 dataset
ifp-0/1/8
  Port state: Uncalibrated
  Port identity: 5C:07:58:A2:CE:5C:00:01:0009
  Port mean path delay: 297
  Log announce interval: -3
  Log sync interval: -4
  Log min delay req interval: -4
  Announce receipt timeout: 3
  Delay mechanism: End to End
  Version number: 2
  Local priority: 128
  Master only: False
  Network interface: ifp-0/1/8
  TTL: 64
  DSCP: 46
  Unicast grant duration: 0 seconds
```

```
    Configured delay asymmetry: 0
    Acceptable master table: Disabled
    Acceptable slave table: Disabled
    Number of foreign masters: 1
    Current foreign master: 0
  Foreign master #0
    L2 address: D8:3A:DD:AA:4D:9D
    GM identity: D8:3A:00:FF:FE:AA:4D:9D
    Port ID: D8:3A:00:FF:FE:AA:4D:9D:0002
    GM clock class: 13
    GM clock accuracy: 0x20
    GM clock offset: 0xFFFF
    GM priority1: 128
    GM priority2: 0
    GM steps removed: 2
    GM sync uncertainty: No
```

#### show ptp event history

You can now run the `show ptp event history` command (with specific filters for grand-master, parent, and interval) to track dataset changes, servo states, and clock transitions. These commands provide real-time visibility into the health of network synchronization and Grandmaster selection directly from the CLI.

**Syntax:**

**show ptp event history**
**show ptp event history interval**
**show ptp event history parent**
**show ptp event history grand-master**

| Attribute | Description |
|---|---|
| `event history` | Displays historical PTP dataset events, including dataset changes, servo states, and clock transitions. |
| `event history interval` | Displays PTP protocol message interval values. |
| `event history parent` | Filters the PTP event history output by parent clock identity, showing dataset events only for the selected parent clock. |
| `event history grand-master` | Filters the PTP event history output by grandmaster clock identity, showing dataset events only for the selected grandmaster clock. |

The example below displays historical PTP dataset events, detailing parent and grandmaster clock identifiers, clock classes, and other synchronization attributes recorded on the system.

```
supervisor@rtbrick: op> show ptp event history
Total number of PTP dataset events: 35
```

```
Dataset change recorded at 2026-06-29 06:55:56
  Protocol event type: ptp_event_protocol_parent_data_set_change
  Parent clock identity: D8:3A:00:FF:FE:AA:4D:9D
  Parent clock port: 2
  Parent stats: No
  Parent offset scaled log variance: Invalid
  Parent clock phase change rate: Invalid
  Grandmaster identity: D8:3A:00:FF:FE:AA:4D:9D
  Grandmaster clock class: 13
  Grandmaster clock accuracy: 0x20
  Grandmaster clock offset: 0xFFFF
  Grandmaster priority 1: 128
  Grandmaster priority 2: 0
Dataset change recorded at 2026-06-29 06:55:53
  Protocol event type: ptp_event_protocol_parent_data_set_change
  Parent clock identity: D8:3A:00:FF:FE:AA:4D:9D
  Parent clock port: 2
  Parent stats: No
  Parent offset scaled log variance: Invalid
  Parent clock phase change rate: Invalid
  Grandmaster identity: D8:3A:00:FF:FE:AA:4D:9D
  Grandmaster clock class: 13
  Grandmaster clock accuracy: 0x21
  Grandmaster clock offset: 0xFFFF
  Grandmaster priority 1: 128
  Grandmaster priority 2: 0

<...>
```

The example below displays historical PTP interval interrupt logs.

```
supervisor@rtbrick: op> show ptp event history interval
Total interval interrupts: 2
Delay Request interrupt recorded at 2026-06-29 06:30:42
  Last interval: 127
  Current interval: -4
Delay Request interrupt recorded at 2026-06-29 06:28:29
  Last interval: 127
  Current interval: -4
```

The example below displays historical changes to the PTP parent data set, documenting three specific events with their associated timestamps, parent clock identities, and port information.

```
supervisor@rtbrick: op> show ptp event history parent
Total records: 3
Parent data set change recorded at 2026-06-29 06:30:39
  Parent clock identity: D8:3A:00:FF:FE:AA:4D:9D
  Parent clock port: 2
  Parent stats: No
```

```
  Parent offset scaled log variance: Invalid
  Parent clock phase change rate: Invalid
Parent data set change recorded at 2026-06-29 06:30:13
  Parent clock identity: 5C:07:58:A2:CE:5C:00:01
  Parent clock port: 0
  Parent stats: No
  Parent offset scaled log variance: Invalid
  Parent clock phase change rate: Invalid
Parent data set change recorded at 2026-06-29 06:28:26
  Parent clock identity: D8:3A:00:FF:FE:AA:4D:9D
  Parent clock port: 2
  Parent stats: No
  Parent offset scaled log variance: Invalid
  Parent clock phase change rate: Invalid
```

The example below displays historical PTP Grandmaster change events, listing their specific timestamps, clock identities, and synchronization attributes.

```
supervisor@rtbrick: op> show ptp event history grand-master
Total records: 35
Grandmaster change recorded at 2026-06-29 06:55:56
  Identity: D8:3A:00:FF:FE:AA:4D:9D
  Clock class: 13
  Clock accuracy: 0x20
  Clock offset: 0xFFFF
  Priority 1: 128
  Priority 2: 0
Grandmaster change recorded at 2026-06-29 06:55:53
  Identity: D8:3A:00:FF:FE:AA:4D:9D
  Clock class: 13
  Clock accuracy: 0x21
  Clock offset: 0xFFFF
  Priority 1: 128
  Priority 2: 0
Grandmaster change recorded at 2026-06-29 06:54:20
  Identity: D8:3A:00:FF:FE:AA:4D:9D
  Clock class: 13
  Clock accuracy: 0x20
  Clock offset: 0xFFFF
  Priority 1: 128
  Priority 2: 0

<...>
```

#### show ptp bmca

You can run **show ptp bmca** and **show ptp bmca interface <interface>** to view critical troubleshooting data, including selected GM clock IDs, clock classes, and selection timelines. These commands provide clearer network visibility into clock selection and synchronization health

directly from the CLI.

**Syntax:**

**show ptp bmca**
**show ptp bmca interface** <interface-name>

| Attribute | Description |
|---|---|
| <cr> | Without any option, it displays all the BMCA details. |
| <interface-name> | Specifies name of the BMCA interface. |

```
supervisor@rtbrick: cfg> show ptp bmca
Selected master clock identity D8:3A:00:FF:FE:AA:4D:9D
  Clock port: 2
  Parent stats: No
  Parent offset scaled log variance: Invalid
  Parent clock phase change rate: Invalid
  Time of selection: 2026-06-29 15:55:05 UTC
  GM Identity: D8:3A:00:FF:FE:AA:4D:9D
  Number of times changed: 6
```

```
supervisor@rtbrick: cfg> show ptp bmca interface ifp-0/1/8
```

### SyncE ESMC Show CLIs

#### show synce statistics

This command displays the SyncE statistics.

**Syntax:**

**show synce statistics** <interface>

| Option | Description |
|---|---|
| <cr> | Without any option, the commands displays the SyncE statistics for all interfaces. |
| <interface> | Displays the SyncE statistics for the specified interface. |

The following example displays the output of the show synce statistics command for all interfaces.

```
supervisor@rtbrick>switch: op> show synce statistics
ifp-0/1/20
    ESMC Rx: 0
```

```
    ESMC Tx: 10719
    Total: 10719
ifp-0/1/69
    ESMC Rx: 0
    ESMC Tx: 30
    Total: 30
```

The following example displays SyncE statistics for the interface.

```
supervisor@rtbrick>switch: op> show synce statistics interface ifp-0/1/20
ifp-0/1/20
    ESMC Rx: 0
    ESMC Tx: 10269
    Total: 10269
```

#### show synce details

This command displays the SyncE details.

**Syntax:**

```
show synce details
```

| Option | Description |
|---|---|
| details | Displays detailed information about the SyncE configurations. |

The following example displays the output of the show synce details command, providing detailed information about the SyncE configurations.

```
supervisor@rtbrick>switch: op> show synce details
Clock-selection: QL-enabled
Equipment clock: EEC-option1
Extended QL-TLV: Disabled
Switchover-mode: Revertive
Hop-count mode: Disable
SyncE clock state: Holdover
DPLL clock state: Holdover
SyncE state duration: 06:33:59
10MHz holdover duration: 10 min
```

#### show synce input-source

This command displays the SyncE input source.

**Syntax:**

```
show synce input-source
```

| Option | Description |
|---|---|
| input-sources | Displays the SyncE input sources. |

The following example displays the output of the show synce input-source command, providing detailed information about the SyncE input sources, including source priority, quality level (QL) status, hold-off timer settings, link status, and clock source ID.

```
supervisor@rtbrick>switch: op> show synce input-source
Interface: ptp
  Source priority: 1
  QL configured: N/A
  QL received: N/A
  QL operational: N/A
  ESMC status: N/A
  Selected source: False
  Hold off timer: 300 ms
  Hold off timer status: False
  WTR timer: 5 mins
  Link status: Down
  WTR timer status: False
  Clock source id: 256
  Signal fail state: True
  Timing loop: False
  Lockout mode: False
```

**show synce interface <interface_name>**

This command displays the SyncE interface information for the specified interface.

**Syntax:**

```
show synce interface <interface_name>
```

| Option | Description |
|---|---|
| <interface_name> | Displays the SyncE interface information for the specified interface. |

The following example displays the output of the `show synce interface ifp-0/1/8` command, providing detailed information about the SyncE status on interface `ifp-0/1/8`. It includes the source type (Output), link status (Up), and quality level (QL) details.

```
supervisor@rtbrick>switch: op> show synce interface ifp-0/1/8
Interface: ifp-0/1/8
```

```
  Source: Output
  Link status: Up
  QL received: N/A
  QL operational: N/A
  QL operational out: QL_DNU
```

#### show synce interface summary

This command displays a summary of the SyncE interface information for all interfaces.

**Syntax:**

```
show synce interface summary
```

This example displays the SyncE interface summary, including details of Interface, QL received, QL operational, Out QL Operational, and Link Status.

```
supervisor@rtbrick>switch: op> show synce interface summary
Interface           QL received    QL operational Out QL Operational  Link Status
ifp-0/1/20          N/A            N/A            QL_SEC              Up
ifp-0/1/69          N/A            N/A            QL_SEC              Up
ifp-0/1/8           N/A            N/A            QL_DNU              Up
ptp                 N/A            N/A            N/A                 Down
```

#### show synce interface <interface-name>

This command displays SyncE interface information for the specified interface.

**Syntax:**

```
show synce interface <interface-name>
```

This example displays the SyncE interface information for the specified interface.

```
supervisor@rtbrick>switch: op> show synce interface interface ifp-0/1/20
Interface           QL received    QL operational Out QL Operational  Link Status
ifp-0/1/20          N/A            N/A            QL_SEC              Up
ifp-0/1/69          N/A            N/A            QL_SEC              Up
ifp-0/1/8           N/A            N/A            QL_DNU              Up
ptp                 N/A            N/A            N/A                 Down
```

#### show synce output-source

This command displays the detailed configuration and status information for the selected SyncE source.

**Syntax:**

```
show synce output-source
```

The following example displays the output of the show synce output-sourcecommand, providing detailed information about the selected SyncE source, including interface name, link status, and quality level (QL) details.

```
supervisor@rtbrick>switch: op> show synce output-source
Interface: ifp-0/1/20
  Link status: Up
  QL configured: N/A
  QL operational: QL_SEC
Interface: ifp-0/1/69
  Link status: Up
  QL configured: N/A
  QL operational: QL_SEC
Interface: ifp-0/1/8
  Link status: Up
  QL configured: QL_DNU
  QL operational: QL_DNU
```

#### show synce selected-source

This command displays detailed view of the selected SyncE source.

**Syntax:**

```
show synce synce selected-source
```

This example provides a detailed view of the selected SyncE source, including key parameters such as Interface, Source Priority, QL (configured, received, and operational), ESMC status, Hold off and WTR timers, Clock source ID, Signal fail state, Timing loop, and Lockout mode.

```
supervisor@rtbrick>switch: op> show synce synce selected-source
Interface: ptp
  Source priority: 1
  QL configured: N/A
  QL received: QL_PRC
  QL operational: QL_PRC
  ESMC status: N/A
  Selected source: True
  Hold off timer: 300 ms
  Hold off timer status: False
  WTR timer: 5 mins
  Link status: Up
```

```
  WTR timer status: False
  Clock source id: 256
  Signal fail state: False
  Timing loop: False
  Lockout mode: False
```

# 11.3. Logging

## 11.3.1. Logging Overview

RBFS logging is the process of writing log messages during the execution of events. RBFS logging provides reports on events across the entire RBFS ecosystem, spanning different functional areas. In a hardware switch, this happens at 2 levels:

- Inside the RBFS container
- In the RtBrick Host

All these logs, generated by different components, can be exported to the log management server, where you can view and analyze real-time data.

RBFS provides logging for the entire RBFS ecosystem, including Brick Daemons (BD), as well as other (non-BD) daemons. Brick Daemons are built on top of Brick Data Store (BDS), and other (non-BD) daemons (such as Prometheus) are the ones that are not dependent on BDS. BDS provides an in-built infrastructure for logging, which can be used by all BDS applications.

The diagram below shows how logs generated within the RBFS container (from Brick Daemon, Prometheus, and Syslog) and from API Gateway Daemon (APIGWD) are sent to Controller Daemon (CTRLD) on the RtBrick host. CTRLD aggregates these logs and exports them to an external Log Management Server for centralized monitoring and analysis.

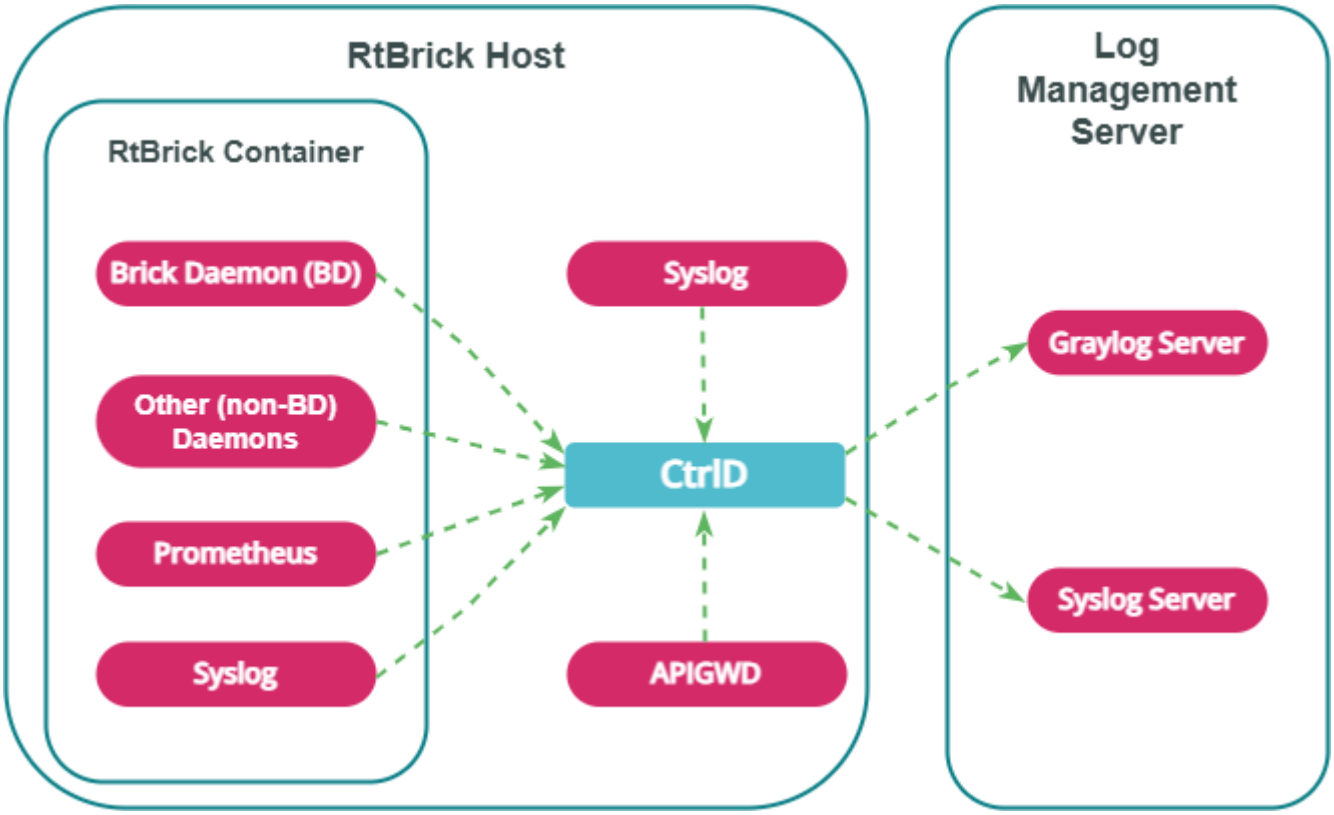



*Figure 164. Logging Overview*

### Logging in RBFS Container

The RBFS container logging infrastructure (BDS Logging) provides low-latency, in-memory logging for large-scale systems without significant performance impact.

#### BDS Logging

#### Log Tables

BDS logs are stored in a BDS table. BDS creates a log table for each module in a BD. One entry is added to this log table for every log. Older entries are removed from the table when the number of entries exceeds 10,000. A log table is created for a module only if that module has at least one logged event. Every module in RBFS has at least one log table named in this format: **<modulename>.<bd-name>.log**.

#### Log Maps

Each log is mapped to a single event logged by the application. To optimize memory usage, RBFS does not store verbose strings; instead, it stores the log map as an identifier to the actual string message.

The log map and the log ID refer to the same entity.

#### Log Modules

Every BDS application consists of multiple modules. Logging can be configured for each BDS module separately.

The log modules supported by RBFS are: access_service, bcm_q2c, bd, bds, bds_mock, bgp, fib, fwdinfra, hostconfd, ifm, igmp, ipoe, isis, l2tp, lag, ldp, license, lldp, lwip_contrib, mrib, nat, oam, ospf, pim, policy, pool, poold, pppoe, pubsub, resmon, rib, rtbrick-cli, secure_management, snapshot, static, subsMgmt, time_series, validation, vpp.

#### Log Groups

A log group is a collection of log maps or log IDs. Groups have been introduced to simplify the log configuration tasks. For example, to debug a BGP peer issue, instead of enabling the entire bgp module logs, you can enable logging for the BGP peer log group.

#### Guidelines and Limitations for BDS Logging

The following are some guidelines and limitations for BDS logging:

- By default, BDS logging is enabled, and the log level is set to 'Error'.
- By default, logging for the BDS and PUBSUB modules has been disabled.
- The user can configure log levels per BD or per module.
- In a scaled setup, logging should not be kept enabled for an extended period.
- When you set the log-level to "None", the log is disabled for the specific module, group, or globally.

#### Non-BDS Logging

#### Syslog

Syslog is generated by the Linux API-based logging mechanism. RBFS Linux container uses syslog for logging. Syslog can also be exported to Graylog. Currently, RBFS supports exporting syslog messages from the Linux system facilities, such as auth, authpriv, daemon, and kern, to Graylog.

#### TSDB (Prometheus) Alert Messages

Prometheus is the system and service monitoring application that can be deployed in RBFS to collect and process metrics. In RBFS, Prometheus-generated alert messages are forwarded to CtrlD, and from there they can be exported to the configured log management servers.

### Logging in to the RtBrick Host

In RBFS, there are daemons that are not part of the RBFS container but run on the RtBrick host. RBFS provides logging for these daemons.

#### CTRLD

CtrlD logging provides log messages of events related to elements, ZTP, and security.

#### APIGWD

ApiGwD logs contain details about who accessed the API and how they accessed it.

### Plugin Alias

Any logs in RBFS can be exported to an external logging destination. Currently, CtrlD supports GELF and syslog as external plugins.

CtrlD is the egress node for all the GELF (Graylog Extended Log Format) messages. The brick daemons are configured to send GELF messages to CtrlD, which forwards them to the configured endpoints, such as syslog or a GELF endpoint. This is because CtrlD enhances the GELF message with switch-global settings (such as the switch's serial number).

### Supported Logging Formats

RBFS supports two types of logging formats: GELF and Syslog.

#### GELF

GELF is a logging format for Graylog. It uses JSON to structure log messages, making them searchable and more structured. The supported protocol for GELF is HTTP.

The diagram below shows the example of a GELF message on the Graylog Web Interface.

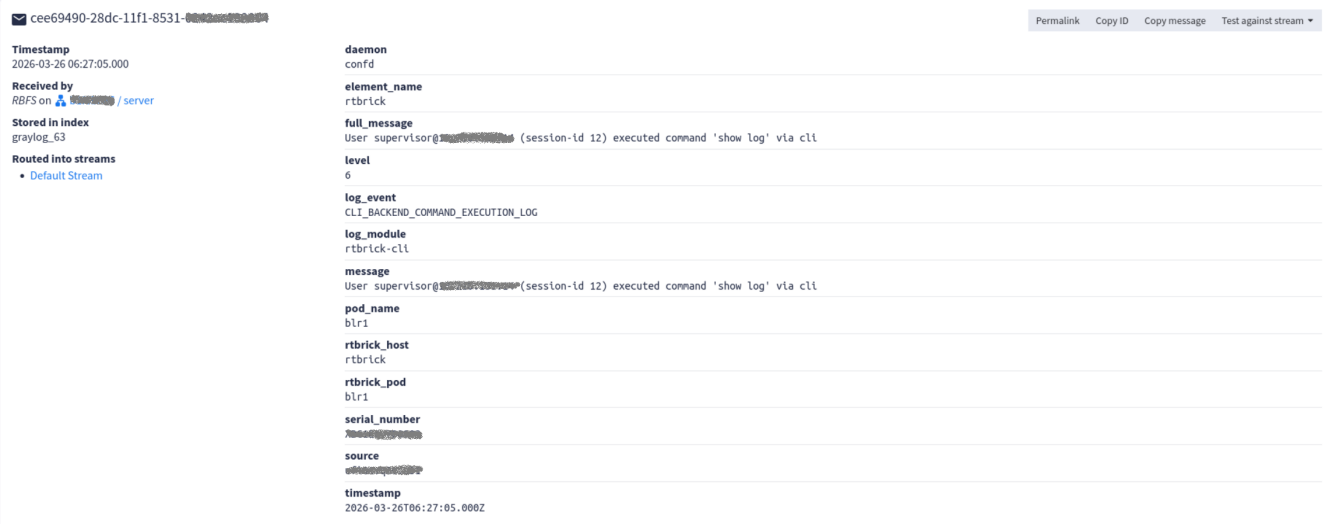



*Figure 165. GELF Message Shown on the Graylog Web Interface*

#### Syslog

Syslog is a format used by Linux systems and many networking devices for logging. It is a plain text protocol with a defined structure. The supported Syslog transport protocols are TCP and UDP.

The diagram below shows the example of a message in syslog format as displayed on the Graylog Web Interface.

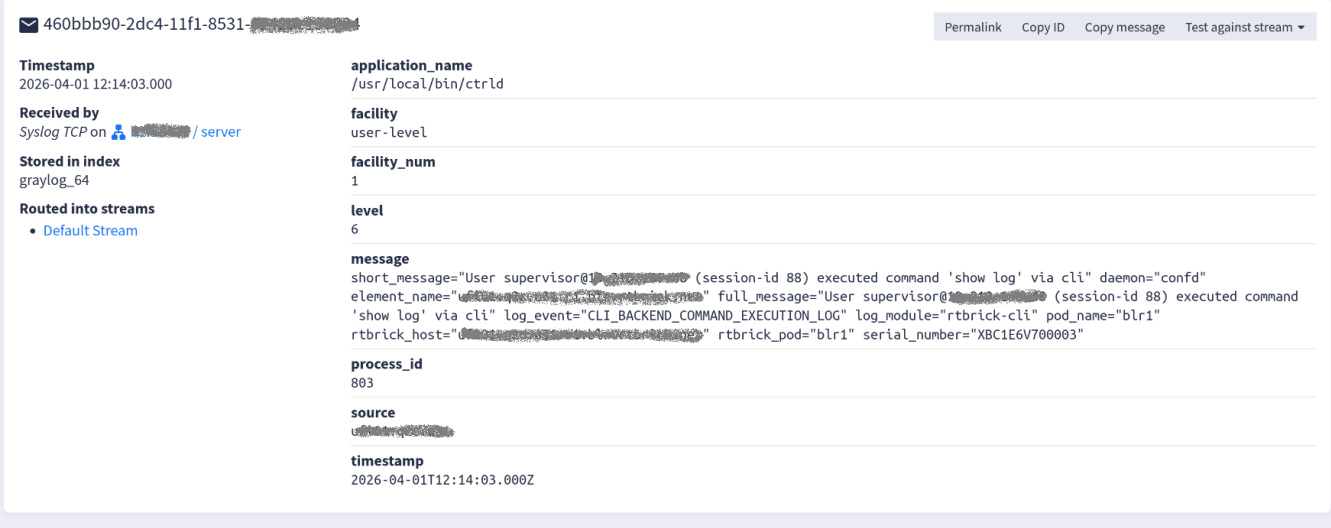



*Figure 166. Syslog Message Format Shown on the Graylog Web Interface*

### Log Levels

Log level indicates the severity of events (logs). You can configure logging by severity level. The diagram below shows the log levels supported by RBFS in order of priority:

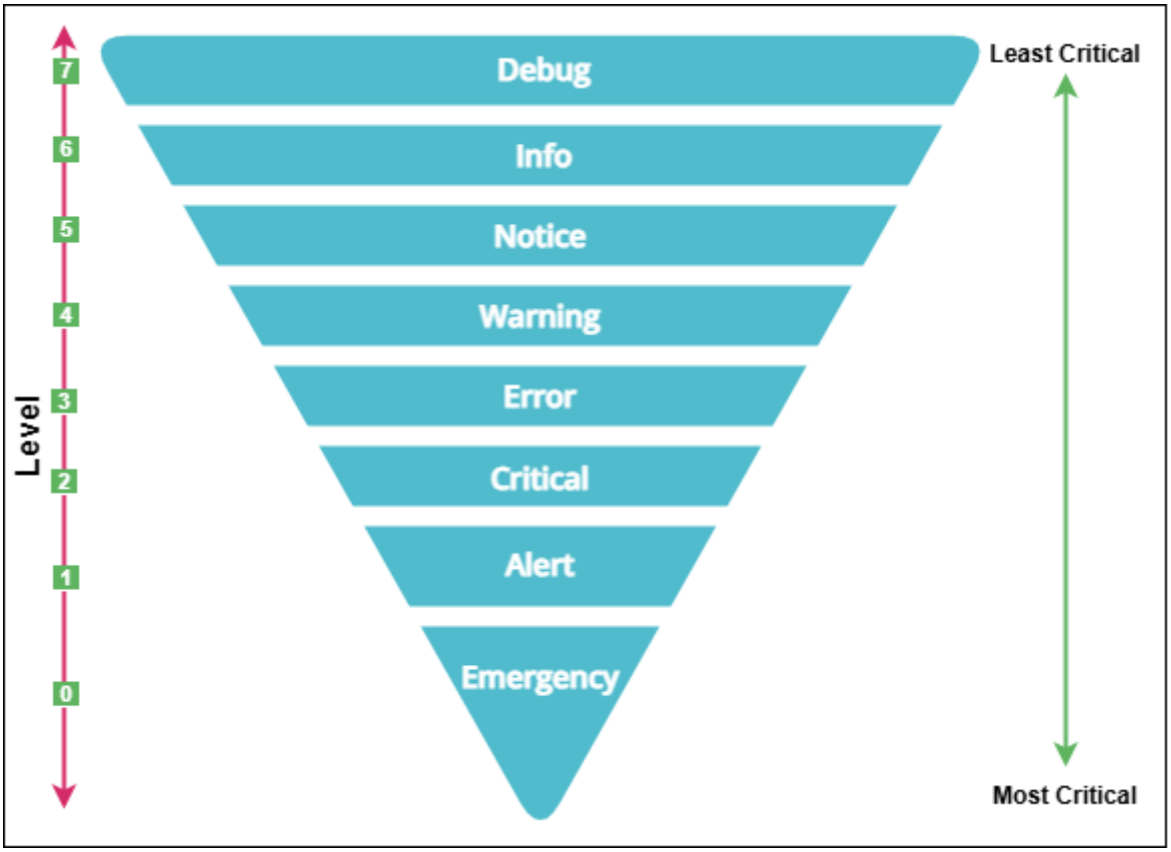



*Figure 167. Log Levels*

| Level | Level Name | Description |
|---|---|---|
| 0 | Emergency | Events that cause the application to terminate. |

| Level | Level Name | Description |
|---|---|---|
| 1 | Alert | A condition that should be corrected immediately. For example, “ISIS_AUTHENTICATION_FAIL” indicates an IS-IS authentication failure. |
| 2 | Critical | Severe error that will prevent the application from continuing. For example, “LICENSE_EXPIRY_CRITICAL” indicates that the current license has expired. |
| 3 | Error | Events that cause issues. For example, “LICENSE_EXPIRY_ERROR” indicates that the current license is about to expire. |
| 4 | Warning | Events that might cause issues. For example, “LICENSE_EXPIRY_WARNING” indicates that the current license is about to expire. |
| 5 | Notice | Conditions that are not error conditions, but that may require special handling. For example, “PPPOE_SESSION_LIMIT_REACHED” indicates that the PPPoE session limit has been reached. |
| 6 | Info | Events showing normal application behaviour. For example, “PPPOE_SESSION_DELETE” indicates that the PPPoE session deletion has occurred. |
| 7 | Debug | Diagnostic information used for debugging. For example, “PPPOE_LCP_START_ECHO” indicates that the PPP LCP echo request interval has started. |

- All log levels below the specified level are logged. For example, if the level is "Warning” (4), logs at levels such as "Error" (3), "Critical" (2), "Alert" (1), and "Emergency" (1) are included.
- Any level above “Warning” indicates that you should log with caution, as they may cause system instability in a scaled environment.

**Supported Platforms**

Not all features are necessarily supported on each hardware platform. Refer to the Platform Guide for the features and sub-features supported by each platform.

## 11.3.2. Logging Configuration

**Configuration Overview**

You can enable logging for:

- Brick Daemon (such as pppod, ipod, confd)
- Modules (such as OSPF, BGP, IS-IS)
- RtBrick Host system

### Configuration Hierarchy

The diagram illustrates the logging configuration hierarchy.

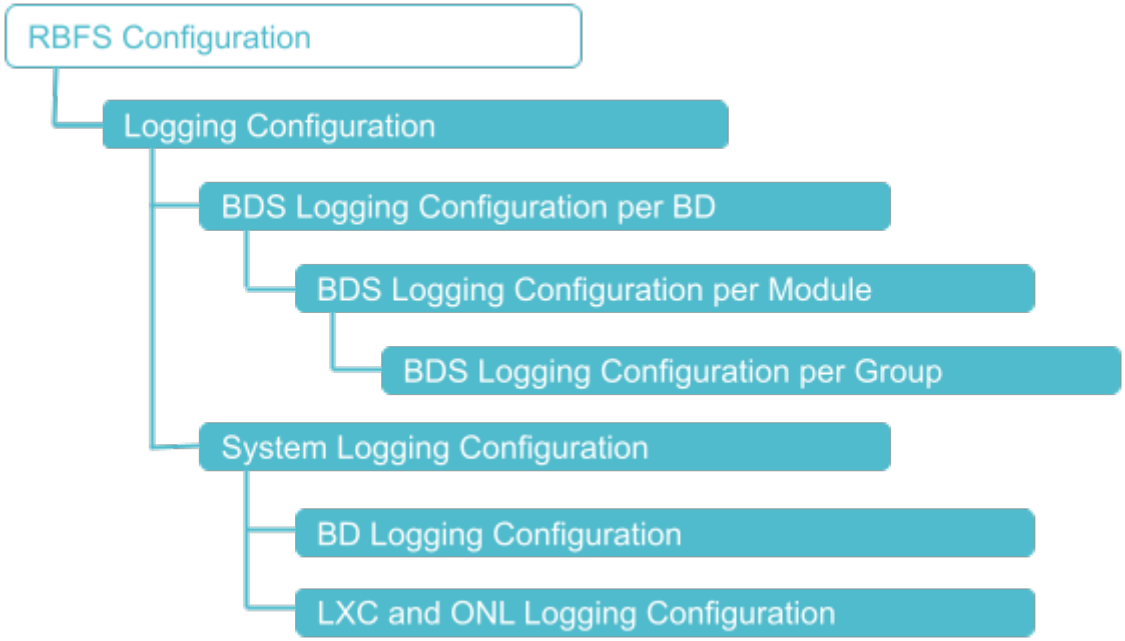



*Figure 168. Logging Configuration Hierarchy*

### Configuration Syntax and Commands

The following sections describe the logging configuration syntax and commands.

#### BDS Logging Configuration

You can configure BDS logging for a BD, a module, and for a group.

> A specific configuration takes priority over a generic configuration. For example, if you have configured a daemon log level of `bgp.iod.1` to "warning", and you have configured a log level of BGP module to "notice", then the final log level of BGP will be "notice".

#### Configuring BDS Logging for a Brick Daemon

You can configure BDS logging for a brick daemon or all daemons to generate logs related that daemon/daemons.

**Syntax:**

```
set log bd <bd-name> <attribute> <value>
```

| Attribute | Description |
|---|---|
| `<bd-name>` | Configure for the specified BD name. |
| `all` | Configure for all BDs. |
| `module <module-name>` | Module name. For more information, see the section "Configuring BDS Logging for a Module". |
| `plugin-alias alias-name <alias-name>` | Specify the external log management server name for sending the logs to the server. |
| `plugin-alias level <level>` | Log severity level. You can filter logs based on the log severity levels for sending to the external log management server. This allows to send only the required log messages to the log management server instead of sending a whole lot of data.<br><br>For example, if you configure `Warning` as the severity level, logs with the severity level 'Warning' and all the log levels above `Warning` such as `Error`, `Critical`, `Alert`, and `Emergency` will be sent to the log management server. |

The following example shows how to configure BDS logging for a brick daemon.

```
set log bd bgp.iod.1 level info
set log bd bgp.iod.1 plugin-alias alias-name ztp plugin-alias level error
```

```
{
  "ietf-restconf:data": {
    "rtbrick-config:system": {},
    "rtbrick-config:log": {
      "bd": [
        {
          "bd-name": "bgp.iod.1",
          "level": "info",
          "plugin-alias": {
            "alias-name": "ztp",
            "level": "error"
          }
        }
      ]
    }
  }
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Configuring BDS Logging for a Module

You can enable logging for a module such as BGP, IS-IS.

**Syntax:**

**set log module** <module-name> <attribute> <value>

| Attribute | Description |
|---|---|
| `<module-name>` | Module name |
| `group <group-name>` | BDS log module log-group configuration. For more information, see the section "Configuring BDS Logging for a Group". |
| `level <level>` | Log severity level. |
| `plugin-alias alias-name <alias-name>` | Plugin alias name |
| `plugin-alias level <level>` | Log level. You can filter logs based on the log severity levels for sending to the log management server. This will help you to send only the required log messages to the log management server instead of sending a whole lot of data.<br><br>For example, if you configure `Warning` as the severity level, logs with the severity level 'Warning' and all the log levels above `Warning` such as `Error`, `Critical`, `Alert`, and `Emergency` will be sent to the log management server. |

The following example shows how to configure BDS logging for a module.

```
set log module igmp level info
set log module igmp plugin-alias alias-name graylog-srv1 plugin-alias level warning
```

```
{
  "ietf-restconf:data": {
    "rtbrick-config:system": {},
    "rtbrick-config:log": {
      "module": [
        {
          "module-name": "igmp",
          "level": "info",
          "plugin-alias": {
            "alias-name": "graylog-srv1",
            "level": "warning"
          }
        }
      ]
    }
  }
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Configuring BDS Logging for a Group

You can enable logging for a group of log IDs. This configuration allows you to generate logs for a group of log IDs (also known as log maps). Using log groups, you can enable logging for a specific set of log IDs. This approach gives you more targeted control over log generation, helping you collect only the relevant information without overwhelming the system.

**Syntax:**

**set log module** <module-name> **group** <group-name> <attribute> <value>

| Attribute | Description |
|---|---|
| `<group-name>` | Group name |
| `level <level>` | Specifies the level of the plug-in alias. |
| `rate-limit <rate-limit>` | Rate-limiting is only supported for log groups. Configuring a higher rate-limit for a whole module may cause system instability due to generation of high volume of logs. The default value is 10. |
| `plugin-alias alias-name <alias-name>` | Plugin alias name |
| `plugin-alias level <level>` | Specifies the level of the plug-in alias. You can filter logs by severity level before sending them to the log management server. This will help you to send only the required log messages to the log management server instead of sending a whole lot of data.<br><br>For example, if you configure `Warning` as the severity level, logs with the severity level 'Warning' and all the log levels above `Warning` such as, `Error`, `Critical`, `Alert`, and `Emergency`, will be sent to the log management server. |

The following example shows how to configure BDS logging for a group.

```
set log module bgp group interface level warning
```

```
{
  "ietf-restconf:data": {
    "rtbrick-config:log": {
      "module": [
        {
          "module-name": "bgp",
          "group": [
            {
              "group-name": "interface",
```

```
              "level": "warning"
            }
          ]
        }
      ]
    }
  }
}
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### System Logging (syslog) Configuration

##### LXC and RtBick Host Logging Configuration

In addition to configuring logging for the various daemons and modules in the RBFS container, you can also enable logging for the underlying RtBrick Host system. You can send logs from Linux and RtBrick Host system facilities, such as `auth`, `authpriv`, `daemon`, and `kern` to the log management server.

Also, you can configure an external log management server to transport logs. Currently, Syslog and Graylog are the log management servers supported by RBFS.

**Syntax:**

**set log system facility** <facility-name> **plugin-alias** <attribute> <value>

| Attribute | Description |
|---|---|
| `<facility-name>` | Supported facilities include `all`, `auth`, `authpriv`, `daemon`, and `kern`. |
| `plugin-alias alias-name <alias-name>` | Plugin alias name |
| `plugin-alias level <level>` | Specify the level of the plug-in alias. You can filter logs based on the log severity levels for sending to the log management server. This will help you to send only the required log messages to the log management server instead of sending a whole lot of data.<br><br>For example, if you configure `Warning` as the severity level, logs with the severity level 'Warning' and all the log levels above `Warning` such as, `Error`, `Critical`, `Alert`, and `Emergency`, will be sent to the log management server. |

The following example shows how to configure system logging for the `kern` facility and send the logs to Graylog.

```
set log system facility kern plugin-alias alias-name graylog plugin-alias level notice
```

```
{
  "ietf-restconf:data": {
    "rtbrick-config:log": {
      "system": {
        "facility": [
          {
            "facility-name": "kern",
            "plugin-alias": {
              "alias-name": "graylog",
              "level": "notice"
            }
          }
        ]
      }
    }
  }
}
```

To access the RESTCONF API that corresponds to this CLI, click here.

#### Configuring an External Log Server (Plugin Alias)

You can configure an external log management server as a destination to send logs. This is done by configuring a plugin alias in RBFS. A plugin alias represents the log management server endpoint to which the logs will be sent. When you configure a plugin alias, you specify the log management server details such as the address, type (Syslog or Graylog), network protocol (TCP or UDP), and severity level of the logs to be sent.

> By default, logs are sent to the log management server through the device management interface (eth0 in hardware devices), which is intended for out-of-band management. If you want to send logs through an in-band connection, you need to enable in-band management on RBFS. For more information, see the section Configuring an External Log Server (Plugin Alias) Using In-band Connection

**Syntax:**

**set system host log-alias** <attribute> <value>

| Attribute | Description |
|---|---|
| `<name>` | Log alias name |
| `<address>` | Plugin address. For example:<br><br>GELF: http://192.0.2.0:1102/gelf<br>Syslog: 198.51.100.0:525 |
| `level` | Severity level for the log. |

| Attribute | Description |
|---|---|
| network | Log alias endpoint port. |
| type | Server type of log alias. |
| allow <regex-pattern> | Allows regular expression pattern used to permit log messages. Only messages matching this pattern are sent to log endpoint (Address) configured. |
| disallow <regex-pattern> | Disallows a regular expression pattern to drop log messages. Any message matching this pattern will be discarded. If empty, no messages are explicitly disallowed. |

The following example shows how to configure a plugin alias for a log management server.

```
set system host log-alias bng endpoint http://192.0.2.0:1102/gelf
set system host log-alias bng endpoint http://192.0.2.0:1102/gelf type gelf
set system host log-alias bng endpoint http://192.0.2.0:1102/gelf network http
set system host log-alias bng endpoint http://192.0.2.0:1102/gelf level info
```

```
{
  "ietf-restconf:data": {
    "rtbrick-config:system": {
      "host": {
        "log-alias": [
          {
            "name": "bng",
            "endpoint": [
              {
                "address": "http://192.0.2.0:1102/gelf",
                "type": "gelf",
                "network": "http",
                "level": "info"
              }
            ]
          }
        ]
      }
    }
  }
}
```

The following example shows how to configure a plugin alias for a Graylog server with the address http://192.0.2.0:1102/gelf:

```
set system host log-alias graylog endpoint http://192.0.2.0:1102/gelf
set system host log-alias graylog endpoint http://192.0.2.0:1102/gelf type gelf
set system host log-alias graylog endpoint http://192.0.2.0:1102/gelf network http
```

```
set system host log-alias graylog endpoint http://192.0.2.0:1102/gelf level info
```

```
{
  "ietf-restconf:data": {
    "rtbrick-config:system": {
      "host": {
        "log-alias": [
          {
            "name": "graylog",
            "endpoint": [
              {
                "address": "http://192.0.2.0:1102/gelf",
                "type": "gelf",
                "network": "http",
                "level": "info"
              }
            ]
          }
        ]
      }
    }
  }
}
```

The following example shows how to configure a plugin alias for a Syslog server using TCP as the transport protocol with the address 198.51.100.0:525:

```
set system host log-alias syslog endpoint 198.51.100.0:525
set system host log-alias syslog endpoint 198.51.100.0:525 type syslog
set system host log-alias syslog endpoint 198.51.100.0:525 network tcp
```

```
supervisor@rtbrick: cfg> show config
{
  "ietf-restconf:data": {
    "rtbrick-config:system": {
      "host": {
        "log-alias": [
          {
            "name": "syslog",
            "endpoint": [
              {
                "address": "198.51.100.0:525",
                "type": "syslog",
                "network": "tcp"
              }
            ]
          }
```

```
        ]
      }
    }
  }
}
```

The following example shows how to configure a plugin alias for a Syslog server using UDP as the transport protocol with the address `198.51.100.0:525`:

```
set system host log-alias syslog endpoint 198.51.100.0:525
set system host log-alias syslog endpoint 198.51.100.0:525 type syslog
set system host log-alias syslog endpoint 198.51.100.0:525 network udp
```

```
supervisor@rtbrick: cfg> show config
{
  "ietf-restconf:data": {
    "rtbrick-config:system": {
      "host": {
        "log-alias": [
          {
            "name": "syslog",
            "endpoint": [
              {
                "address": "198.51.100.0:525",
                "type": "syslog",
                "network": "udp"
              }
            ]
          }
        ]
      }
    }
  }
}
```

The example configuration below sets up an external log server with the alias "graylog" and applies filters to allow only messages related to BGP and OSPF while disallowing specific BGP message patterns. Additionally, it configures another log alias "syslog" to send all debug-level messages related to BGP down, flap, or Idle events to a syslog server.

```
set system host log-alias graylog
set system host log-alias graylog endpoint http://192.0.2.0:1102/gelf
set system host log-alias graylog endpoint http://192.0.2.0:1102/gelf type gelf
set system host log-alias graylog endpoint http://192.0.2.0:1102/gelf network http
set system host log-alias graylog endpoint http://192.0.2.0:1102/gelf level info
set system host log-alias graylog endpoint http://192.0.2.0:1102/gelf allow
"(BGP|OSPF)"
```

```
set system host log-alias graylog endpoint http://192.0.2.0:1102/gelf disallow "(BGP
message received|BGP update message)"
set system host log-alias syslogset system host log-alias syslog endpoint
198.51.100.0:525
set system host log-alias syslog endpoint 198.51.100.0:525 type syslog
set system host log-alias syslog endpoint 198.51.100.0:525 network tcp
set system host log-alias syslog endpoint 198.51.100.0:525 level debug
set system host log-alias syslog endpoint 198.51.100.0:525 allow
"BGP.*(down|flap|Idle)"
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Configuring an External Log Server (Plugin Alias) Using In-band Connection

By default, plugins send messages exclusively through the device management interface (eth0 in hardware devices), which is intended for out-of-band management. However, you can also configure an external log server (Plugin Alias) using an in-band connection. This requires in-band to be enabled on RBFS.

**Syntax:**

**set inband-management instance** <instance-name> **connection-tracking true**

```
set inband-management instance default connection-tracking true
```

The external log server's IP address must be reachable from the in-band network.

The following example shows how to configure a plugin alias for a Syslog server using UDP as the transport protocol with the address `198.51.100.0:525` via in-band connection:

```
set system host log-alias syslog endpoint 198.51.100.0:525
set system host log-alias syslog endpoint 198.51.100.0:525 type syslog
set system host log-alias syslog endpoint 198.51.100.0:525 network udp
set inband-management instance default connection-tracking true
```

```
supervisor@rtbrick: cfg> show config
{
  "ietf-restconf:data": {
    "rtbrick-config:system": {
      "load-last-config": "true",
      "host": {
        "log-alias": [
          {
            "name": "syslog",
            "endpoint": [
```

```
              {
                "address": "198.51.100.0:525",
                "type": "syslog",
                "network": "udp"
              }
            ]
          }
        ]
      }
    },
    "rtbrick-config:inband-management": {
      "instance": [
        {
          "name": "default",
          "connection-tracking": "true"
        }
      ]
    },
```

#### Enabling Terminal Logging

Terminal logging captures all input and output from command-line sessions, including logs from a brick daemon (BD) that are streamed to the operator's terminal and stored in the system log table. When terminal logging is enabled for a particular bd or all bds, the logs are displayed on the terminal whenever they are generated.

**Syntax:**

**terminal-logging enable bd** <bd_name>
**terminal-logging disable bd** <bd_name>
**terminal-logging enable all**
**terminal-logging disable all**

| Attribute | Description |
|---|---|
| `enable bd <bd_name>` | Enables terminal logging for the specified brick daemon. |
| `disable bd <bd_name>` | Disables terminal logging for the specified brick daemon. |
| `enable all` | Enables terminal logging for all brick daemons. |
| `disable all` | Disables terminal logging for all brick daemons. |

The example below enables terminal logging for all BDs.

```
supervisor@rtbrick.net: cfg> terminal-logging enable all
Terminal logging enabled for all BD
[   Notice   ] <2026-07-09T11:27:25.308043+0000> User supervisor@192.168.1.1 (session-
id 408) executed command 'terminal-logging enable all' via cli
```

The details of the terminal log are stored in log tables, which can be viewed using the "show log" command.

```
supervisor@rtbrick.net: cfg> show log
[  Notice  ] <2026-07-09T11:15:59.231464+0000> Launch service rtbrick-ribd, PID 357
[  Notice  ] <2026-07-09T11:15:59.233600+0000> Launch service rtbrick-staticd, PID 381
[  Notice  ] <2026-07-09T11:15:59.240945+0000> Launch service rtbrick-lldpd, PID 352
[  Notice  ] <2026-07-09T11:15:59.323574+0000> Launch service rtbrick-mribd, PID 355
[  Notice  ] <2026-07-09T11:15:59.351000+0000> Launch service rtbrick-ifmd, PID 373
[  Notice  ] <2026-07-09T11:15:59.425134+0000> Launch service rtbrick-fibd, PID 401
[  Notice  ] <2026-07-09T11:17:01.471547+0000> Launch service rtbrick-ptpd, PID 687
[   Info   ] <2026-07-09T11:27:44.718447+0000> Admin status of the physical interface
ifp-0/1/69 is DOWN
[   Info   ] <2026-07-09T11:27:44.720037+0000> Link status of the physical interface
ifp-0/1/69 is UP
[   Info   ] <2026-07-09T11:27:44.720085+0000> Operational status of the physical
interface ifp-0/1/69 is DOWN
[   Info   ] <2026-07-09T11:27:46.742890+0000> Admin status of the physical interface
ifp-0/1/69 is DOWN
<...>
```

## 11.3.3. Logging Operational Commands

The logging operational commands provide information about the logging operations. They are used to show logs in the system, log configuration status and so on.

### BDS Logging

The BDS logging show commands provide information about the BDS logging operations. With the BDS logging operational commands, you can verify BDS logging configuration status and view BDS logs.

#### Verifying BDS Logging Configuration Status

##### Show Log Status

This command shows log configuration status for all modules except infrastructure modules in all BDs. The default `show log status` displays the summary of log status for the whole system and there are options available to show specific module or BD log status.

**Syntax:**

**show log status** <attribute> <value>

| Option | Description |
|---|---|
| - | Without any option, the command displays the log configuration status for all modules except infrastructure modules in all BDs. |

| Option | Description |
| --- | --- |
| `bd <bd-name>` | Displays log status of the all modules including infrastructure in the specified BD. |
| `detail` | Displays the log configuration status in detail all the way till log IDs. |
| `module <module-name>` | Displays the log status for a given module in all BDs where this module is running. |

Example: Summary of log status for all modules

```
supervisor@rtbrick: op> show log status
Module log status:
  access_service:
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        general                  error          None                      none
10
        generic                  error          None                      none
10
  bcm_q2c:
    fibd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        bds                      error          None                      none
10
        config                   error          None                      none
10
        ecmp                     error          None                      none
10
        general                  error          None                      none
10
        generic                  error          None                      none
10
        ifl                      error          None                      none
10
        l2tp                     error          None                      none
10
        l2x                      error          None                      none
10
        li                       error          None                      none
10
        port                     error          None                      none
10
```

```
        pppoe                     error          None                      none
10
        qos                       error          None                      none
10
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                     Level          Plugin                    Plugin Level
Rate limit
        bds                       error          None                      none
10
        config                    error          None                      none
10
        ecmp                      error          None                      none
10
        general                   error          None                      none
10
        generic                   error          None                      none
10
        ifl                       error          None                      none
10
        l2tp                      error          None                      none
10
        l2x                       error          None                      none
10
        li                        error          None                      none
10
        port                      error          None                      none
10
        pppoe                     error          None                      none
10
        qos                       error          None                      none
10
  bds_mock:
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                     Level          Plugin                    Plugin Level
Rate limit
        generic                   error          None                      none
10
  bgp:
    bgp.iod.1:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                     Level          Plugin                    Plugin Level
Rate limit
        config                    error          None                      none
10
        flowspec                  error          None                      none
10
```

```
        general                  error          None                      none
10
        generic                  error          None                      none
10
        instance                 error          None                      none
10
        interface                error          None                      none
10
        message                  error          None                      none
10
        peer                     error          None                      none
10
        validation               error          None                      none
10
    bgp.appd.1:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        config                   error          None                      none
10
        flowspec                 error          None                      none
10
        general                  error          None                      none
10
        generic                  error          None                      none
10
        instance                 error          None                      none
10
        interface                error          None                      none
10
        message                  error          None                      none
10
        peer                     error          None                      none
10
        validation               error          None                      none
10
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        config                   error          None                      none
10
        flowspec                 error          None                      none
10
        general                  error          None                      none
10
        generic                  error          None                      none
10
        instance                 error          None                      none
```

```
10
        interface                error          None                      none
10
        message                  error          None                      none
10
        peer                     error          None                      none
10
        validation               error          None                      none
10
  fib:
    fibd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        adjacency                error          None                      none
10
        bds                      error          None                      none
10
        config                   error          None                      none
10
        general                  error          None                      none
10
        generic                  error          None                      none
10
        instance-afi-safi        error          None                      none
10
        interface-events         error          None                      none
10
        route                    error          None                      none
10
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        adjacency                error          None                      none
10
        bds                      error          None                      none
10
        config                   error          None                      none
10
        general                  error          None                      none
10
        generic                  error          None                      none
10
        instance-afi-safi        error          None                      none
10
        interface-events         error          None                      none
10
        route                    error          None                      none
```

```
10
  fwdinfra:
    fibd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        generic                  error          None                      none
10
    ifmd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        generic                  error          None                      none
10
    mribd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        generic                  error          None                      none
10
    ribd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        generic                  error          None                      none
10
    staticd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        generic                  error          None                      none
10
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        generic                  error          None                      none
10
  hostconfd:
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        bds                      error          None                      none
```

```
10
        config                  error          None                     none
10
        functional              error          None                     none
10
        generic                 error          None                     none
10
  ifm:
    ifmd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                   Level          Plugin                   Plugin Level
Rate limit
        bds                     error          None                     none
10
        config                  error          None                     none
10
        general                 error          None                     none
10
        generic                 error          None                     none
10
        interface-events        error          None                     none
10
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                   Level          Plugin                   Plugin Level
Rate limit
        bds                     error          None                     none
10
        config                  error          None                     none
10
        general                 error          None                     none
10
        generic                 error          None                     none
10
        interface-events        error          None                     none
10
  igmp:
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                   Level          Plugin                   Plugin Level
Rate limit
        config                  error          None                     none
10
        generic                 error          None                     none
10
        igmp-interface-events   error          None                     none
10
        igmp-membership-events  error          None                     none
```

```
10
        igmp-packet-events       error          None                     none
10
        igmp-route-events        error          None                     none
10
  ipoe:
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level
Rate limit
        config                   error          None                     none
10
        dhcp                     error          None                     none
10
        dhcp-relay               error          None                     none
10
        general                  error          None                     none
10
        generic                  error          None                     none
10
        redundancy               error          None                     none
10
        subscriber               error          None                     none
10
  isis:
    isis.iod.1:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level
Rate limit
        config                   error          None                     none
10
        general                  error          None                     none
10
        generic                  error          None                     none
10
        instance                 error          None                     none
10
        interface                error          None                     none
10
        lsp                      error          None                     none
10
        message                  error          None                     none
10
        neighbor                 error          None                     none
10
        route                    error          None                     none
10
        spf                      error          None                     none
10
```

```
    isis.appd.1:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        config                   error          None                      none
10
        general                  error          None                      none
10
        generic                  error          None                      none
10
        instance                 error          None                      none
10
        interface                error          None                      none
10
        lsp                      error          None                      none
10
        message                  error          None                      none
10
        neighbor                 error          None                      none
10
        route                    error          None                      none
10
        spf                      error          None                      none
10
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        config                   error          None                      none
10
        general                  error          None                      none
10
        generic                  error          None                      none
10
        instance                 error          None                      none
10
        interface                error          None                      none
10
        lsp                      error          None                      none
10
        message                  error          None                      none
10
        neighbor                 error          None                      none
10
        route                    error          None                      none
10
        spf                      error          None                      none
10
  l2tp:
```

```
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        config                   error          None                      none
10
        general                  error          None                      none
10
        generic                  error          None                      none
10
        session                  error          None                      none
10
        subscriber               error          None                      none
10
        tunnel                   error          None                      none
10
  lag:
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        generic                  error          None                      none
10
  ldp:
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        adjacency                error          None                      none
10
        generic                  error          None                      none
10
        notification             error          None                      none
10
        peer                     error          None                      none
10
  lldp:
    lldpd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        bds                      error          None                      none
10
        config                   error          None                      none
10
        ems                      error          None                      none
10
```

```
        general                  error          None                     none
10
        generic                  error          None                     none
10
        interface                error          None                     none
10
        neighbor                 error          None                     none
10
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level
Rate limit
        bds                      error          None                     none
10
        config                   error          None                     none
10
        ems                      error          None                     none
10
        general                  error          None                     none
10
        generic                  error          None                     none
10
        interface                error          None                     none
10
        neighbor                 error          None                     none
10
  mrib:
    mribd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level
Rate limit
        generic                  error          None                     none
10
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level
Rate limit
        generic                  error          None                     none
10
  nat:
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level
Rate limit
        bds                      error          None                     none
10
        general                  error          None                     none
```

```
10
        generic                  error          None                      none
10
        nat-resource             error          None                      none
10
  oam:
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        bfd                      error          None                      none
10
        cfm                      error          None                      none
10
        config                   error          None                      none
10
        fsm                      error          None                      none
10
        general                  error          None                      none
10
        generic                  error          None                      none
10
        mpls-oam                 error          None                      none
10
        session                  error          None                      none
10
    oamd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        bfd                      error          None                      none
10
        cfm                      error          None                      none
10
        config                   error          None                      none
10
        fsm                      error          None                      none
10
        general                  error          None                      none
10
        generic                  error          None                      none
10
        mpls-oam                 error          None                      none
10
        session                  error          None                      none
10
  ospf:
    ospf.appd.1:
      Level: error, Plugin: None, Plugin Level: none
```

```
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        generic                  error          None                      none
10
    ospf.iod.1:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        generic                  error          None                      none
10
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        generic                  error          None                      none
10
  pim:
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        dr-election              error          None                      none
10
        error-log                error          None                      none
10
        generic                  error          None                      none
10
        info-log                 error          None                      none
10
        instance                 error          None                      none
10
        interface                error          None                      none
10
        join-prune               error          None                      none
10
        multicast-route          error          None                      none
10
        neighbor                 error          None                      none
10
        object-trace             error          None                      none
10
        packet                   error          None                      none
10
        rpf                      error          None                      none
10
  policy:
    policy.server:
```

```
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        BDS                      error          None                      none
10
        Configuration            error          None                      none
10
        Enforcement              error          None                      none
10
        Generation               error          None                      none
10
        generic                  error          None                      none
10
    ospf.appd.1:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        BDS                      error          None                      none
10
        Configuration            error          None                      none
10
        Enforcement              error          None                      none
10
        Generation               error          None                      none
10
        generic                  error          None                      none
10
    isis.iod.1:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        BDS                      error          None                      none
10
        Configuration            error          None                      none
10
        Enforcement              error          None                      none
10
        Generation               error          None                      none
10
        generic                  error          None                      none
10
    ospf.iod.1:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        BDS                      error          None                      none
10
```

```
        Configuration            error          None                     none
10
        Enforcement              error          None                     none
10
        Generation               error          None                     none
10
        generic                  error          None                     none
10
    bgp.iod.1:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level
Rate limit
        BDS                      error          None                     none
10
        Configuration            error          None                     none
10
        Enforcement              error          None                     none
10
        Generation               error          None                     none
10
        generic                  error          None                     none
10
    bgp.appd.1:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level
Rate limit
        BDS                      error          None                     none
10
        Configuration            error          None                     none
10
        Enforcement              error          None                     none
10
        Generation               error          None                     none
10
        generic                  error          None                     none
10
    isis.appd.1:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level
Rate limit
        BDS                      error          None                     none
10
        Configuration            error          None                     none
10
        Enforcement              error          None                     none
10
        Generation               error          None                     none
10
```

```
        generic                  error          None                     none
10
    mribd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level
Rate limit
        BDS                      error          None                     none
10
        Configuration            error          None                     none
10
        Enforcement              error          None                     none
10
        Generation               error          None                     none
10
        generic                  error          None                     none
10
    lldpd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level
Rate limit
        BDS                      error          None                     none
10
        Configuration            error          None                     none
10
        Enforcement              error          None                     none
10
        Generation               error          None                     none
10
        generic                  error          None                     none
10
    ribd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level
Rate limit
        BDS                      error          None                     none
10
        Configuration            error          None                     none
10
        Enforcement              error          None                     none
10
        Generation               error          None                     none
10
        generic                  error          None                     none
10
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level
```

```
Rate limit
        BDS                     error          None                     none
10
        Configuration           error          None                     none
10
        Enforcement             error          None                     none
10
        Generation              error          None                     none
10
        generic                 error          None                     none
10
  poold:
    poold:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                   Level          Plugin                   Plugin Level
Rate limit
        generic                 error          None                     none
10
  pppoe:
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                   Level          Plugin                   Plugin Level
Rate limit
        config                  error          None                     none
10
        dhcpv6                  error          None                     none
10
        general                 error          None                     none
10
        generic                 error          None                     none
10
        ppp                     error          None                     none
10
        pppoe                   error          None                     none
10
        subscriber              error          None                     none
10
  resmon:
    resmond:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                   Level          Plugin                   Plugin Level
Rate limit
        generic                 error          None                     none
10
        resmon-bds-events       error          None                     none
10
        resmon-internal-events  error          None                     none
10
```

```
        resmon-ntp-peer-events   error          None                     none
10
        resmon-resource-events   error          None                     none
10
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level
Rate limit
        generic                  error          None                     none
10
        resmon-bds-events        error          None                     none
10
        resmon-internal-events   error          None                     none
10
        resmon-ntp-peer-events   error          None                     none
10
        resmon-resource-events   error          None                     none
10
  rib:
    mribd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level
Rate limit
        bds                      error          None                     none
10
        config                   error          None                     none
10
        general                  error          None                     none
10
        generic                  error          None                     none
10
        neighbor-events          error          None                     none
10
        routing                  error          None                     none
10
    ribd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level
Rate limit
        bds                      error          None                     none
10
        config                   error          None                     none
10
        general                  error          None                     none
10
        generic                  error          None                     none
10
        neighbor-events          error          None                     none
```

```
10
        routing                  error          None                      none
10
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        bds                      error          None                      none
10
        config                   error          None                      none
10
        general                  error          None                      none
10
        generic                  error          None                      none
10
        neighbor-events          error          None                      none
10
        routing                  error          None                      none
10
  rtbrick-cli:
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        cli-execution            info           None                      none
0
        generic                  error          None                      none
10
  static:
    staticd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        bds                      error          None                      none
10
        config                   error          None                      none
10
        general                  error          None                      none
10
        generic                  error          None                      none
10
        static-fib-local         error          None                      none
10
        static-rib-local         error          None                      none
10
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
```

```
        Group                    Level          Plugin                    Plugin Level
Rate limit
        bds                      error          None                      none
10
        config                   error          None                      none
10
        general                  error          None                      none
10
        generic                  error          None                      none
10
        static-fib-local         error          None                      none
10
        static-rib-local         error          None                      none
10
  subsMgmt:
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        bds                      error          None                      none
10
        config                   error          None                      none
10
        general                  error          None                      none
10
        generic                  error          None                      none
10
        l2bsa                    error          None                      none
10
        lawful-intercept         error          None                      none
10
        pool                     error          None                      none
10
        radius                   error          None                      none
10
        redundancy               error          None                      none
10
        subscriber               error          None                      none
10
        subscriberfsm            error          None                      none
10
  validation:
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        config                   error          None                      none
10
        general                  error          None                      none
```

```
10
        generic                 error          None                     none
10
        rpki                    error          None                     none
10
  vpp:
    fibd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                   Level          Plugin                   Plugin Level
Rate limit
        adjacency               error          None                     none
10
        bds                     error          None                     none
10
        config                  error          None                     none
10
        general                 error          None                     none
10
        generic                 error          None                     none
10
        instance-afi-safi       error          None                     none
10
        interface-events        error          None                     none
10
        neighbor                error          None                     none
10
        route                   error          None                     none
10
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                   Level          Plugin                   Plugin Level
Rate limit
        adjacency               error          None                     none
10
        bds                     error          None                     none
10
        config                  error          None                     none
10
        general                 error          None                     none
10
        generic                 error          None                     none
10
        instance-afi-safi       error          None                     none
10
        interface-events        error          None                     none
10
        neighbor                error          None                     none
10
        route                   error          None                     none
```

```
10
```

Example: Summary of log status for the specified module

```
supervisor@rtbrick: op> show log status module bgp
Module log status:
  bgp:
    bgp.iod.1:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        config                   error          None                      none
10
        flowspec                 error          None                      none
10
        general                  error          None                      none
10
        generic                  error          None                      none
10
        instance                 error          None                      none
10
        interface                error          None                      none
10
        message                  error          None                      none
10
        peer                     error          None                      none
10
        validation               error          None                      none
10
    bgp.appd.1:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        config                   error          None                      none
10
        flowspec                 error          None                      none
10
        general                  error          None                      none
10
        generic                  error          None                      none
10
        instance                 error          None                      none
10
        interface                error          None                      none
10
        message                  error          None                      none
10
        peer                     error          None                      none
```

```
10
        validation               error          None                      none
10
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                    Plugin Level
Rate limit
        config                   error          None                      none
10
        flowspec                 error          None                      none
10
        general                  error          None                      none
10
        generic                  error          None                      none
10
        instance                 error          None                      none
10
        interface                error          None                      none
10
        message                  error          None                      none
10
        peer                     error          None                      none
10
        validation               error          None                      none
10
```

Example: Log status of all modules including infrastructure in the specified BD.

```
supervisor@rtbrick: op> show log status bd bgp.appd.1
System/File log status:
  Level: notice

Module log status:
  bd:
    Level: notice, Plugin: None, Plugin Level: none
    Log group status:
      Group                    Level          Plugin                    Level
Rate limit
      dns-resolver             notice         None                      none
10
      generic                  notice         None                      none
10
      http                     notice         None                      none
10
      service-launch           notice         None                      none
10
      udp                      notice         None                      none
10
  bds:
```

```
    Level: none, Plugin: None, Plugin Level: none
    Log group status:
      Group                    Level          Plugin                    Level
Rate limit
      generic                  none           None                      none
10
      object                   none           None                      none
10
      table                    none           None                      none
10
      trim                     none           None                      none
10
  bgp:
    Level: error, Plugin: None, Plugin Level: none
    Log group status:
      Group                    Level          Plugin                    Level
Rate limit
      config                   error          None                      none
10
      flowspec                 error          None                      none
10
      general                  error          None                      none
10
      generic                  error          None                      none
10
      instance                 error          None                      none
10
      interface                error          None                      none
10
      message                  error          None                      none
10
      peer                     error          None                      none
10
      validation               error          None                      none
10
  license:
    Level: error, Plugin: None, Plugin Level: none
    Log group status:
      Group                    Level          Plugin                    Level
Rate limit
      generic                  error          None                      none
10
      internal                 error          None                      none
10
      operational              error          None                      none
10
  lwip_contrib:
    Level: error, Plugin: None, Plugin Level: none
    Log group status:
      Group                    Level          Plugin                    Level
Rate limit
```

```
      generic                  error          None                      none
10
  policy:
    Level: error, Plugin: None, Plugin Level: none
    Log group status:
      Group                    Level          Plugin                    Level
Rate limit
      BDS                      error          None                      none
10
      Configuration            error          None                      none
10
      Enforcement              error          None                      none
10
      Generation               error          None                      none
10
      generic                  error          None                      none
10
  pool:
    Level: error, Plugin: None, Plugin Level: none
    Log group status:
      Group                    Level          Plugin                    Level
Rate limit
      generic                  error          None                      none
10
      internal                 error          None                      none
10
      operational              error          None                      none
10
  pubsub:
    Level: none, Plugin: None, Plugin Level: none
    Log group status:
      Group                    Level          Plugin                    Level
Rate limit
      generic                  none           None                      none
10
  secure_management:
    Level: error, Plugin: None, Plugin Level: none
    Log group status:
      Group                    Level          Plugin                    Level
Rate limit
      general                  error          None                      none
10
      generic                  error          None                      none
10
      session-context          info           None                      none
0
  snapshot:
    Level: error, Plugin: None, Plugin Level: none
    Log group status:
      Group                    Level          Plugin                    Level
Rate limit
```

```
      generic                  error          None                      none
10
  time_series:
    Level: error, Plugin: None, Plugin Level: none
    Log group status:
      Group                    Level          Plugin                    Level
Rate limit
      generic                  error          None                      none
10
```

Example: Log status for the specified module in the specified BD

```
supervisor@rtbrick: cfg>  show log status module bgp bd bgp.appd.1
Module log status:
  bgp:
    bgp.appd.1:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin         Plugin Level   Rate limit
        config                   error          None           none           10
        flowspec                 error          None           none           10
        general                  error          None           none           10
        generic                  error          None           none           10
        instance                 error          None           none           10
        interface                error          None           none           10
        message                  error          None           none           10
        peer                     error          None           none           10
        validation               error          None           none           10
```

Example: Detailed Log status for the specified module in the specified BD

```
supervisor@rtbrick: op> show log status bd bgp.appd.1 detail
System/File log status:
  Level: notice

Module log status:
  bd:
    Level: notice, Plugin: None, Plugin Level: none
    Log group status:
      dns-resolver, Level: notice, Plugin: None, Plugin Level: none, Rate limit: 10
     Log ID status:
        LOG ID                                                              Level
Plugin                   Level          Rate limit
        BD_UV_DNS_TEMPLATE_MISSING_CB                                       notice
None                     none           10
        BD_UV_GETNAMEINFO_FAILED                                            notice
None                     none           10
      generic, Level: notice, Plugin: None, Plugin Level: none, Rate limit: 10
      http, Level: notice, Plugin: None, Plugin Level: none, Rate limit: 10
```

```
     Log ID status:
        LOG ID                                                          Level
Plugin                    Level          Rate limit
        HTTP_CLIENT_REQUEST_LOG                                         notice
None                      none           10
        HTTP_JWK_FILE_JSON_PARSE_FAILED                                 notice
None                      none           10
        HTTP_JWK_FILE_MEM_ALLOC_FAILED                                  notice
None                      none           10
        HTTP_JWK_FILE_MISSING                                           notice
None                      none           10
        HTTP_JWK_FILE_OPEN_FAILED                                       notice
None                      none           10
        HTTP_JWK_FILE_READ_FAILED                                       notice
None                      none           10
        HTTP_JWK_MISSING_KEY                                            notice
None                      none           10
        HTTP_JWK_MULTIPLE_KEYS                                          notice
None                      none           10
        HTTP_RESP_INSUFFICIENT_BUFFER                                   notice
None                      none           10
        HTTP_SEND                                                       notice
None                      none           10
        HTTP_WRITE_BUFFER_MEM_ALLOC_FAILED                              notice
None                      none           10
        HTTP_WRITE_PRINTF_FAILED                                        notice
None                      none           10
      service-launch, Level: notice, Plugin: None, Plugin Level: none, Rate limit: 10
     Log ID status:
        LOG ID                                                          Level
Plugin                    Level          Rate limit
        BD_SERVICE_RESTART                                              notice
None                      none           10
        BD_SERVICE_RESTART_FAILED                                       notice
None                      none           10
        BD_SERVICE_START                                                notice
None                      none           10
        BD_SERVICE_START_FAILED                                         notice
None                      none           10
        BD_SERVICE_START_REQUEST                                        notice
None                      none           10
        BD_SERVICE_STOP                                                 notice
None                      none           10
        BD_SERVICE_STOP_FAILED                                          notice
None                      none           10
        BD_SERVICE_STOP_REQUEST                                         notice
None                      none           10
      udp, Level: notice, Plugin: None, Plugin Level: none, Rate limit: 10
     Log ID status:
        LOG ID                                                          Level
Plugin                    Level          Rate limit
```

```
        BD_UV_UDP_BIND_FAILED                                                   notice
None                    none           10
        BD_UV_UDP_CONNECT_FAILED                                                notice
None                    none           10
        BD_UV_UDP_INVALID_ADDR_FAMILY                                           notice
None                    none           10
        BD_UV_UDP_MISSING_CB                                                    notice
None                    none           10
        BD_UV_UDP_READ_START_FAILED                                             notice
None                    none           10
  bds:
    Level: none, Plugin: None, Plugin Level: none
    Log group status:
      generic, Level: none, Plugin: None, Plugin Level: none, Rate limit: 10
     Log ID status:
        LOG ID                                                                  Level
Plugin                  Level          Rate limit
        BDS_ATTRIBUTE_TEMPLATE_EVENT                                            none
None                    none           10
        BDS_INVALID_PARAMS                                                      none
None                    none           10
        BDS_PUBSUB_ERROR_STATUS                                                 none
None                    none           10
        BDS_QUEUE_TABLE                                                         none
None                    none           10
        BDS_TEST_LOG                                                            none
None                    none           10
      object, Level: none, Plugin: None, Plugin Level: none, Rate limit: 10
     Log ID status:
        LOG ID                                                                  Level
Plugin                  Level          Rate limit
        BDS_INVALID_OBJECT_TEMPLATE                                             none
None                    none           10
        BDS_INVALID_OBJECT_TEMPLATE_ATTRIBUTE                                   none
None                    none           10
        BDS_OBJECT_ATTRIBUTE_EVENT                                              none
None                    none           10
        BDS_OBJECT_EVENT                                                        none
None                    none           10
        BDS_OBJECT_TEMPLATE_EVENTS                                              none
None                    none           10
        BDS_OBJECT_TEMPLATE_GET_ERROR                                           none
None                    none           10
        BDS_ROOT_OBJECT_EVENT                                                   none
None                    none           10
      table, Level: none, Plugin: None, Plugin Level: none, Rate limit: 10
     Log ID status:
        LOG ID                                                                  Level
Plugin                  Level          Rate limit
        BDS_TABLE_CHANGE_CALLBACK                                               none
None                    none           10
```

```
        BDS_TABLE_ERROR_STATUS                                                  none
None                    none           10
        BDS_TABLE_EVENT                                                         none
None                    none           10
        BDS_TABLE_EVENTS                                                        none
None                    none           10
        BDS_TABLE_SEQ_BLOCK                                                     none
None                    none           10
      trim, Level: none, Plugin: None, Plugin Level: none, Rate limit: 10
     Log ID status:
        LOG ID                                                                  Level
Plugin                  Level          Rate limit
        BDS_TRIM_COUNT                                                          none
None                    none           10
        BDS_TRIM_ENTRY_ADD_FAILED                                               none
None                    none           10
        BDS_TRIM_INFRA_TABLE_ADD_FAILED                                         none
None                    none           10
        BDS_TRIM_NOT_SUPPORTED                                                  none
None                    none           10
        BDS_TRIM_OBJECT_DELETE_FAILED                                           none
None                    none           10
  bgp:
    Level: error, Plugin: None, Plugin Level: none
    Log group status:
      config, Level: error, Plugin: None, Plugin Level: none, Rate limit: 10
      flowspec, Level: error, Plugin: None, Plugin Level: none, Rate limit: 10
     Log ID status:
        LOG ID                                                                  Level
Plugin                  Level          Rate limit
        BGP_FLOWSPEC_ACL_ADDED                                                  error
None                    none           10
        BGP_FLOWSPEC_ACL_DELETED                                                error
None                    none           10
        BGP_FLOWSPEC_ACL_REINSTALLED                                            error
None                    none           10
        BGP_FLOWSPEC_ENTRY_DELETED                                              error
None                    none           10
        BGP_FLOWSPEC_PRIORITY_ADDED                                             error
None                    none           10
        BGP_FLOWSPEC_PRIORITY_DELETED                                           error
None                    none           10
        BGP_FLOWSPEC_PRIORITY_FOUND                                             error
None                    none           10
        BGP_FLOWSPEC_PRIORITY_FULL                                              error
None                    none           10
        BGP_FLOWSPEC_PRIORITY_NOT_FOUND                                         error
None                    none           10
        BGP_FLOWSPEC_PRIORITY_UPDATED                                           error
None                    none           10
        BGP_FLOWSPEC_RECEIVED                                                   error
```

```
None                    none          10
        BGP_FLOWSPEC_RULE_INVALID                                          error
None                    none          10
        BGP_FLOWSPEC_SEND                                                  error
None                    none          10
        BGP_FLOWSPEC_STATUS_INVALID                                        error
None                    none          10
      general, Level: error, Plugin: None, Plugin Level: none, Rate limit: 10
      generic, Level: error, Plugin: None, Plugin Level: none, Rate limit: 10
     Log ID status:
        LOG ID                                                             Level
Plugin                  Level         Rate limit
        BGP_DETAIL_ADDR_RESOLVE_UPDATE                                     error
None                    none          10
        BGP_DETAIL_AFI_SAFI_CFG_ERROR                                      error
None                    none          10
        BGP_DETAIL_ALL_AS_REMOVED                                          error
None                    none          10
        BGP_DETAIL_BEST_ROUTE_JOB_START                                    error
None                    none          10
        BGP_DETAIL_BEST_ROUTE_JOB_STOP                                     error
None                    none          10
        BGP_DETAIL_CFG_SRGB                                                error
None                    none          10
        BGP_DETAIL_COMM_NO_ADVERT                                          error
None                    none          10
        BGP_DETAIL_COMM_NO_EXPORT                                          error
None                    none          10
        BGP_DETAIL_CONNECT_COLLUSION                                       error
None                    none          10
        BGP_DETAIL_CONN_RETRY_TMR_EXP                                      error
None                    none          10
        BGP_DETAIL_FWDD_FIB_WALK_JOB                                       error
None                    none          10
        BGP_DETAIL_GC_TMR_EXP                                              error
None                    none          10
        BGP_DETAIL_GLOBAL_INST_ERROR                                       error
None                    none          10
        BGP_DETAIL_HOLD_TMR_EXP                                            error
None                    none          10
        BGP_DETAIL_IFL_CHANGE_FAIL                                         error
None                    none          10
        BGP_DETAIL_IFL_PEER_PROCESSING                                     error
None                    none          10
        BGP_DETAIL_IMPORT_RT_ERROR                                         error
None                    none          10
        BGP_DETAIL_INIT_SERVER                                             error
None                    none          10
        BGP_DETAIL_INSTANCE_ACTION                                         error
None                    none          10
        BGP_DETAIL_INSTANCE_INACTIVE                                       error
```

```
None                    none          10
        BGP_DETAIL_JOB_COMPLETE                                          error
None                    none          10
        BGP_DETAIL_JOB_PAUSED                                            error
None                    none          10
        BGP_DETAIL_JOB_START                                             error
None                    none          10
        BGP_DETAIL_LOCAL_FIB_WALK_JOB                                    error
None                    none          10
        BGP_DETAIL_MAX_AS_PATHS                                          error
None                    none          10
        BGP_DETAIL_MAX_CLUSTER_LIST                                      error
None                    none          10
        BGP_DETAIL_NHOP_LABEL_NO_ROUTE                                   error
None                    none          10
        BGP_DETAIL_NH_RESOLVE_JOB                                        error
None                    none          10
        BGP_DETAIL_NO_NH_SELF_FOR_LL_PEER                                error
None                    none          10
        BGP_DETAIL_NO_SERVICE_LABEL                                      error
None                    none          10
        BGP_DETAIL_NO_SRGB                                               error
None                    none          10
        BGP_DETAIL_OPTION_B_LABEL                                        error
None                    none          10
        BGP_DETAIL_PEER_CFG_UPD_EXP                                      error
None                    none          10
        BGP_DETAIL_PEER_DEACTIVATED                                      error
None                    none          10
        BGP_DETAIL_PEER_IGNORED_NO_IFA                                   error
None                    none          10
        BGP_DETAIL_PEER_ROUTE_COUNT_EXP                                  error
None                    none          10
        BGP_DETAIL_PEER_START_TMR_EXP                                    error
None                    none          10
        BGP_DETAIL_PEER_TIMER_START                                      error
None                    none          10
        BGP_DETAIL_PEER_TIMER_STOP                                       error
None                    none          10
        BGP_DETAIL_PG_RESTART                                            error
None                    none          10
        BGP_DETAIL_PG_WALK_JOB                                           error
None                    none          10
        BGP_DETAIL_POLICY_BLOCK                                          error
None                    none          10
        BGP_DETAIL_POLICY_DENY                                           error
None                    none          10
        BGP_DETAIL_POLICY_FAIL                                           error
None                    none          10
        BGP_DETAIL_POOL_OBJ                                              error
None                    none          10
```

```
        BGP_DETAIL_POOL_RELEASE                                                 error
None                    none          10
        BGP_DETAIL_POOL_REQ_ERR                                                 error
None                    none          10
        BGP_DETAIL_POOL_RES_ERR                                                 error
None                    none          10
        BGP_DETAIL_POOL_RES_REPLY                                               error
None                    none          10
        BGP_DETAIL_RD_ERROR                                                     error
None                    none          10
        BGP_DETAIL_REDIST_INSTANCE                                              error
None                    none          10
        BGP_DETAIL_REDIS_IGNORED                                                error
None                    none          10
        BGP_DETAIL_REGISTER_TABLE                                               error
None                    none          10
        BGP_DETAIL_REMOVE_PRIVATE_AS                                            error
None                    none          10
        BGP_DETAIL_RIBIN_OPTB_WALK                                              error
None                    none          10
        BGP_DETAIL_RIBIN_WALK_JOB                                               error
None                    none          10
        BGP_DETAIL_ROUTE_RT_ERROR                                               error
None                    none          10
        BGP_DETAIL_RR_DISABLED                                                  error
None                    none          10
        BGP_DETAIL_SERVICE_LABEL                                                error
None                    none          10
        BGP_DETAIL_SID_OVERFLOW                                                 error
None                    none          10
        BGP_DETAIL_TABLE_CREATED                                                error
None                    none          10
        BGP_DETAIL_TABLE_DELETED                                                error
None                    none          10
        BGP_DETAIL_TCP_ACCEPT                                                   error
None                    none          10
        BGP_DETAIL_TCP_CLOSE                                                    error
None                    none          10
        BGP_DETAIL_TCP_CONNECT                                                  error
None                    none          10
        BGP_DETAIL_TCP_ERROR                                                    error
None                    none          10
        BGP_DETAIL_UPD_AS_LOOP                                                  error
None                    none          10
        BGP_DETAIL_UPD_CLUSTER_LOOP                                             error
None                    none          10
        BGP_DETAIL_UPD_PG_ADD                                                   error
None                    none          10
        BGP_DETAIL_UPD_PG_DEL                                                   error
None                    none          10
        BGP_DETAIL_UPD_PG_FSM_CHANGE                                            error
```

```
None                    none          10
        BGP_DETAIL_UPD_PG_REFRESH                                       error
None                    none          10
        BGP_DETAIL_VRF_AF_CONFIG                                        error
None                    none          10
        BGP_ERROR_ADDR_RESOLUTION_REQ                                   error
None                    none          10
        BGP_ERROR_ADDR_RES_REQ_DELETE                                   error
None                    none          10
        BGP_ERROR_AFI_SAFI_CODE                                         error
None                    none          10
        BGP_ERROR_AFI_SAFI_INDEX                                        error
None                    none          10
        BGP_ERROR_ARRAY_TEMPLATE                                        error
None                    none          10
        BGP_ERROR_BD_NAME                                               error
None                    none          10
        BGP_ERROR_BGP_MSG_VALIDATION                                    error
None                    none          10
        BGP_ERROR_CLUSTER_ID_NOT_FOUND                                  error
None                    none          10
        BGP_ERROR_COPY_TEMPLATE                                         error
None                    none          10
        BGP_ERROR_CREATING_NH_ARRAY                                     error
None                    none          10
        BGP_ERROR_CREATING_NH_OBJECT                                    error
None                    none          10
        BGP_ERROR_CREATING_NH_SET                                       error
None                    none          10
        BGP_ERROR_DEFAULT_IF                                            error
None                    none          10
        BGP_ERROR_ENCODE                                                error
None                    none          10
        BGP_ERROR_EXTRACT_OBJ                                           error
None                    none          10
        BGP_ERROR_EXTRACT_OBJECT_TEMPLATE                               error
None                    none          10
        BGP_ERROR_FIND_ADDR_RES_REQ                                     error
None                    none          10
        BGP_ERROR_FIND_OPTB_LABEL                                       error
None                    none          10
        BGP_ERROR_FIND_SERVICE_LABEL                                    error
None                    none          10
        BGP_ERROR_FLUSH_TABLE                                           error
None                    none          10
        BGP_ERROR_GET_ARRAY_VALUE                                       error
None                    none          10
        BGP_ERROR_GET_FIB_NH_DEL                                        error
None                    none          10
        BGP_ERROR_GET_GLOBAL_CONFIG                                     error
None                    none          10
```

```
        BGP_ERROR_GET_OBJECT                                                    error
None                    none           10
        BGP_ERROR_HDL_INSTANCE                                                  error
None                    none           10
        BGP_ERROR_IFA                                                           error
None                    none           10
        BGP_ERROR_INHERIT                                                       error
None                    none           10
        BGP_ERROR_INSTANCE_NAME                                                 error
None                    none           10
        BGP_ERROR_INTERNAL                                                      error
None                    none           10
        BGP_ERROR_INVALID_AFI                                                   error
None                    none           10
        BGP_ERROR_INVALID_AFISAFI_IDX                                           error
None                    none           10
        BGP_ERROR_INVALID_ARGUMENTS                                             error
None                    none           10
        BGP_ERROR_INVALID_JOB_TYPE                                              error
None                    none           10
        BGP_ERROR_INVALID_PREFIX_TYPE                                           error
None                    none           10
        BGP_ERROR_INVALID_ROUTE_SRC                                             error
None                    none           10
        BGP_ERROR_INVALID_SAFI                                                  error
None                    none           10
        BGP_ERROR_IP                                                            error
None                    none           10
        BGP_ERROR_JOB_FAIL                                                      error
None                    none           10
        BGP_ERROR_JOB_OBJECT                                                    error
None                    none           10
        BGP_ERROR_LABEL                                                         error
None                    none           10
        BGP_ERROR_LISTEN_SOCKET                                                 error
None                    none           10
        BGP_ERROR_MALLOC_FAIL                                                   error
None                    none           10
        BGP_ERROR_MAX_LABEL                                                     error
None                    none           10
        BGP_ERROR_MISSING_DISCRETIONARY_ATTR                                    error
None                    none           10
        BGP_ERROR_MISSING_MANDATORY_ATTR                                        error
None                    none           10
        BGP_ERROR_NEXTHOP                                                       error
None                    none           10
        BGP_ERROR_NO_AFI_SAFI                                                   error
None                    none           10
        BGP_ERROR_OBJECT_ADD                                                    error
None                    none           10
        BGP_ERROR_OBJECT_DEL                                                    error
```

```
None                    none          10
        BGP_ERROR_OBJECT_TEMPLATE                                       error
None                    none          10
        BGP_ERROR_OPT_B_LABEL_REQ                                       error
None                    none          10
        BGP_ERROR_PATH_ATTRIBUTE_ADD                                    error
None                    none          10
        BGP_ERROR_PEER_FROM_RIB                                         error
None                    none          10
        BGP_ERROR_PEER_NOT_FOUND                                        error
None                    none          10
        BGP_ERROR_PG_AFCFG                                              error
None                    none          10
        BGP_ERROR_PG_AFI_SAFI_CFG                                       error
None                    none          10
        BGP_ERROR_PG_CFG                                                error
None                    none          10
        BGP_ERROR_PG_REMOTE_AS                                          error
None                    none          10
        BGP_ERROR_POOL_RELEASE                                          error
None                    none          10
        BGP_ERROR_POOL_RES_TEMPLATE                                     error
None                    none          10
        BGP_ERROR_RCVD_MSG                                              error
None                    none          10
        BGP_ERROR_SEND_MSG                                              error
None                    none          10
        BGP_ERROR_SERVICE_LABEL_REQ                                     error
None                    none          10
        BGP_ERROR_SERVICE_NH                                            error
None                    none          10
        BGP_ERROR_TABLE_ATTRIB_NOT_FOUND                                error
None                    none          10
        BGP_ERROR_TABLE_CREATION_FAILED                                 error
None                    none          10
        BGP_ERROR_TABLE_DELETION_FAILED                                 error
None                    none          10
        BGP_ERROR_TABLE_NOT_FOUND                                       error
None                    none          10
        BGP_ERROR_TCP_TABLE                                             error
None                    none          10
        BGP_ERROR_TRAP                                                  error
None                    none          10
        BGP_ERROR_UPDATE_AF                                             error
None                    none          10
        BGP_ERROR_UPDATE_LOCAL_AS                                       error
None                    none          10
        BGP_ERROR_UPDATE_LOCAL_ROUTE                                    error
None                    none          10
        BGP_ERROR_UPDATE_RCVD_ROUTE                                     error
None                    none          10
```

```
        BGP_EXTN_ADDR_FAMILY_DISABLED                                   error
None                    none          10
        BGP_EXTN_ADDR_RESOLUTION_NO_NH                                  error
None                    none          10
        BGP_EXTN_ADDR_RESOLUTION_NO_NH_REQ                              error
None                    none          10
        BGP_EXTN_ADDR_RESOLUTION_TABLE                                  error
None                    none          10
        BGP_EXTN_BGP_NOT_IN_IF                                          error
None                    none          10
        BGP_EXTN_BGP_PEER_CLEANUP                                       error
None                    none          10
        BGP_EXTN_BGP_PEER_PG_WALK                                       error
None                    none          10
        BGP_EXTN_BGP_SEND                                               error
None                    none          10
        BGP_EXTN_CFG_SRGB                                               error
None                    none          10
        BGP_EXTN_CREATING_NH_IGNORED                                    error
None                    none          10
        BGP_EXTN_EOR                                                    error
None                    none          10
        BGP_EXTN_FLOWSPEC_DELETE_FAILED                                 error
None                    none          10
        BGP_EXTN_INSTANCE_BD                                            error
None                    none          10
        BGP_EXTN_INSTANCE_BD_UNCONFIG                                   error
None                    none          10
        BGP_EXTN_INSTANCE_ORPHAN                                        error
None                    none          10
        BGP_EXTN_LU_PREFIX_UPDATE                                       error
None                    none          10
        BGP_EXTN_LU_PREFIX_WITHDRAW                                     error
None                    none          10
        BGP_EXTN_NH_CREATION_FAILED                                     error
None                    none          10
        BGP_EXTN_NH_CREATION_SUCCESS                                    error
None                    none          10
        BGP_EXTN_NOT_DEFAULT_BD                                         error
None                    none          10
        BGP_EXTN_NOT_LL_ADDRESS                                         error
None                    none          10
        BGP_EXTN_NO_DEFAULT_BD_MAPPING                                  error
None                    none          10
        BGP_EXTN_NO_IMPORT_RT_MATCH                                     error
None                    none          10
        BGP_EXTN_PREFIX_IGNORED                                         error
None                    none          10
        BGP_EXTN_PREFIX_IGNORED1                                        error
None                    none          10
        BGP_EXTN_PREFIX_NOT_FOUND                                       error
```

```
None                    none          10
        BGP_EXTN_PREFIX_UPDATE                                          error
None                    none          10
        BGP_EXTN_PREFIX_VALIDATION_ERROR                                error
None                    none          10
        BGP_EXTN_PREFIX_WITHDRAW                                        error
None                    none          10
        BGP_EXTN_SEND_ADDR_RESOLUTION_REQ                               error
None                    none          10
        BGP_EXTN_SEND_EOR                                               error
None                    none          10
        BGP_EXTN_SEND_LU_PREFIX_UPDATE                                  error
None                    none          10
        BGP_EXTN_SEND_LU_PREFIX_WITHDRAW                                error
None                    none          10
        BGP_EXTN_SEND_PREFIX_UPDATE                                     error
None                    none          10
        BGP_EXTN_SEND_PREFIX_WITHDRAW                                   error
None                    none          10
        BGP_EXTN_SEND_VPN_PREFIX_UPDATE                                 error
None                    none          10
        BGP_EXTN_SEND_VPN_PREFIX_WITHDRAW                               error
None                    none          10
        BGP_EXTN_TABLE_ADD_PREFIX                                       error
None                    none          10
        BGP_EXTN_TABLE_ADD_PREFIX_ERROR                                 error
None                    none          10
        BGP_EXTN_TABLE_CREATION_CALLBACK                                error
None                    none          10
        BGP_EXTN_TABLE_DELETION_CALLBACK                                error
None                    none          10
        BGP_EXTN_TABLE_DEL_PREFIX                                       error
None                    none          10
        BGP_EXTN_UPD_INVALID_PATH_LEN                                   error
None                    none          10
        BGP_EXTN_UPD_INVALID_SEND_PATH_ID                               error
None                    none          10
        BGP_EXTN_UPD_LOCAL_PREF_EBGP                                    error
None                    none          10
        BGP_EXTN_UPD_MANDATORY_ATTR_SKIP                                error
None                    none          10
        BGP_EXTN_VPN_PREFIX_UPDATE                                      error
None                    none          10
        BGP_EXTN_VPN_PREFIX_WITHDRAW                                    error
None                    none          10
        BGP_LOG_AF_CONFIG                                               error
None                    none          10
        BGP_LOG_APP_INIT                                                error
None                    none          10
        BGP_LOG_BD_CONFIG                                               error
None                    none          10
```

```
        BGP_LOG_BD_CONFIG_CHANGE                                        error
None                    none          10
        BGP_LOG_EXT_NH                                                  error
None                    none          10
        BGP_LOG_GLOBAL_CONFIG                                           error
None                    none          10
        BGP_LOG_INHERIT_FAILURE                                         error
None                    none          10
        BGP_LOG_INSTANCE                                                error
None                    none          10
        BGP_LOG_INSTANCE_NOT_CONFIG                                     error
None                    none          10
        BGP_LOG_INSTANCE_START                                          error
None                    none          10
        BGP_LOG_INSTANCE_STOP                                           error
None                    none          10
        BGP_LOG_MAX_VRF                                                 error
None                    none          10
        BGP_LOG_MSG_INVALID_PARA_TYPE                                   error
None                    none          10
        BGP_LOG_MULTI_LABEL                                             error
None                    none          10
        BGP_LOG_NO_PG                                                   error
None                    none          10
        BGP_LOG_PEER_AF_NOT_NEGOTIATED                                  error
None                    none          10
        BGP_LOG_PEER_CFG_ACTION                                         error
None                    none          10
        BGP_LOG_PEER_CREATE                                             error
None                    none          10
        BGP_LOG_PEER_DELETE                                             error
None                    none          10
        BGP_LOG_PEER_IGNORE_AF                                          error
None                    none          10
        BGP_LOG_PEER_IGNORE_AF_SEND_LABEL                               error
None                    none          10
        BGP_LOG_PEER_INACTIVE                                           error
None                    none          10
        BGP_LOG_PEER_INACTIVE_ACTION                                    error
None                    none          10
        BGP_LOG_PEER_PG_INACTIVE                                        error
None                    none          10
        BGP_LOG_PEER_RR_NOT_NEGOTIATED                                  error
None                    none          10
        BGP_LOG_PEER_STOP_INVALID_NOTIFY                                error
None                    none          10
        BGP_LOG_PG_AF_CONFIG                                            error
None                    none          10
        BGP_LOG_PG_CFG_ACTION                                           error
None                    none          10
        BGP_LOG_PG_CONFIG                                               error
```

```
None                    none          10
        BGP_LOG_REDIST_CONFIG                                           error
None                    none          10
        BGP_LOG_SUB_ATTR_TO_TABLE                                       error
None                    none          10
        BGP_LOG_SUB_TO_TABLE                                            error
None                    none          10
        BGP_LOG_TABLE_CREATE                                            error
None                    none          10
        BGP_LOG_TABLE_DELETE                                            error
None                    none          10
        BGP_LOG_UNSUB_ATTR_TO_TABLE                                     error
None                    none          10
        BGP_LOG_UNSUB_TO_TABLE                                          error
None                    none          10
        BGP_LOG_VRF_AF_CONFIG                                           error
None                    none          10
        BGP_TRACE_LOG                                                   error
None                    none          10
      instance, Level: error, Plugin: None, Plugin Level: none, Rate limit: 10
     Log ID status:
        LOG ID                                                          Level
Plugin                  Level         Rate limit
        BGP_INSTANCE_INACTIVE                                           error
None                    none          10
      interface, Level: error, Plugin: None, Plugin Level: none, Rate limit: 10
     Log ID status:
        LOG ID                                                          Level
Plugin                  Level         Rate limit
        BGP_IFA_CHANGE                                                  error
None                    none          10
        BGP_IFL_CHANGE                                                  error
None                    none          10
      message, Level: error, Plugin: None, Plugin Level: none, Rate limit: 10
     Log ID status:
        LOG ID                                                          Level
Plugin                  Level         Rate limit
        BGP_FSM_PEER_STATE_CHANGE                                       error
None                    none          10
        BGP_MSG_DECODE_FAIL                                             error
None                    none          10
        BGP_MSG_VALIDATION_FAIL                                         error
None                    none          10
        BGP_UPD_ATTR_EXT_LEN                                            error
None                    none          10
        BGP_UPD_ATTR_INVALID_ASPATH_LEN                                 error
None                    none          10
        BGP_UPD_ATTR_INVALID_ORIGIN                                     error
None                    none          10
        BGP_UPD_ATTR_OPTIONAL_NON_TRANS                                 error
None                    none          10
```

```
        BGP_UPD_ATTR_OPTIONAL_TRANS                                     error
None                    none           10
        BGP_UPD_ATTR_REPEAT                                             error
None                    none           10
        BGP_UPD_ATTR_WELLKNOWN_TRANS                                    error
None                    none           10
        BGP_UPD_ATTR_WK_TRANS_PARTIAL                                   error
None                    none           10
        BGP_UPD_ENFORCE_FIRST_AS                                        error
None                    none           10
        BGP_UPD_FLOWSPEC_INVALID_LEN                                    error
None                    none           10
        BGP_UPD_INVALID_ATTR_EBGP                                       error
None                    none           10
        BGP_UPD_INVALID_EXP_LEN                                         error
None                    none           10
        BGP_UPD_INVALID_IP6_PREFIX                                      error
None                    none           10
        BGP_UPD_INVALID_IPV4_PREFIX                                     error
None                    none           10
        BGP_UPD_INVALID_LEN                                             error
None                    none           10
        BGP_UPD_INVALID_NH                                              error
None                    none           10
        BGP_UPD_INVALID_SEG_TYPE                                        error
None                    none           10
        BGP_UPD_INVALID_SRGB_LEN                                        error
None                    none           10
        BGP_UPD_INVALID_SR_IDX_LEN                                      error
None                    none           10
        BGP_UPD_MALFORMED_PATH_ATTR                                     error
None                    none           10
        BGP_UPD_MREACH_FAIL                                             error
None                    none           10
        BGP_UPD_NLRI_FAIL                                               error
None                    none           10
        BGP_UPD_OWN_ROUTER_ID                                           error
None                    none           10
        BGP_UPD_PATH_ATTR_FAIL                                          error
None                    none           10
        BGP_UPD_RCVD_NON_NEGOTIATED_AF                                  error
None                    none           10
        BGP_UPD_SET_ERR_INFO                                            error
None                    none           10
      peer, Level: error, Plugin: None, Plugin Level: none, Rate limit: 10
     Log ID status:
        LOG ID                                                          Level
Plugin                  Level          Rate limit
        BGP_DUPLICATE_TCP_ACCEPT                                        error
None                    none           10
        BGP_IFL_NBR_ADD                                                 error
```

```
None                    none          10
        BGP_IFL_NBR_DEL                                                 error
None                    none          10
        BGP_INVALID_RECEIVED_AS                                         error
None                    none          10
        BGP_MSG_MANDATORY_ATTRIBUTE                                     error
None                    none          10
        BGP_MSG_OPEN_HOLDTIMER                                          error
None                    none          10
        BGP_MSG_OPEN_IDENTIFIER                                         error
None                    none          10
        BGP_MSG_OPEN_VERSION                                            error
None                    none          10
        BGP_NO_LOCAL_AS                                                 error
None                    none          10
        BGP_NO_ROUTER_ID                                                error
None                    none          10
        BGP_PEER_RESET                                                  error
None                    none          10
        BGP_PEER_RESET_FSM_ERR                                          error
None                    none          10
        BGP_PEER_RESET_INVALID_CAP_LEN                                  error
None                    none          10
        BGP_PEER_RESET_INVALID_PARAMETER_TYPE                           error
None                    none          10
        BGP_PEER_RESET_INVALID_TYPE                                     error
None                    none          10
        BGP_PEER_RESET_MSG_HDR_LEN_ERR                                  error
None                    none          10
        BGP_PEER_STATE_CHANGE                                           error
None                    none          10
      validation, Level: error, Plugin: None, Plugin Level: none, Rate limit: 10
     Log ID status:
        LOG ID                                                          Level
Plugin                  Level         Rate limit
        BGP_VALIDATION_AS_PATH_NULL                                     error
None                    none          10
        BGP_VALIDATION_FILTER_DISABLE                                   error
None                    none          10
        BGP_VALIDATION_FOUND_INVALID                                    error
None                    none          10
        BGP_VALIDATION_FOUND_VALID                                      error
None                    none          10
        BGP_VALIDATION_NOT_FOUND                                        error
None                    none          10
        BGP_VALIDATION_POLICY_UPDATE                                    error
None                    none          10
        BGP_VALIDATION_STATE_CHECK                                      error
None                    none          10
        BGP_VALIDATION_TABLE_NOT_FOUND                                  error
None                    none          10
```

```
        BGP_VALIDATION_UPDATE                                                   error
None                    none           10
        BGP_VALIDATION_UPDATE_DONE                                              error
None                    none           10
  license:
    Level: error, Plugin: None, Plugin Level: none
    Log group status:
      generic, Level: error, Plugin: None, Plugin Level: none, Rate limit: 10
      internal, Level: error, Plugin: None, Plugin Level: none, Rate limit: 10
     Log ID status:
        LOG ID                                                                  Level
Plugin                  Level          Rate limit
        LICENSE_CONFIG_TABLE_CREATE_FAILED                                      error
None                    none           10
        LICENSE_DIRECTORY_CLOSE_FAILURE                                         error
None                    none           10
        LICENSE_DIRECTORY_OPEN_FAILURE                                          error
None                    none           10
        LICENSE_END_DATE_MISSING                                                error
None                    none           10
        LICENSE_FILE_OPEN_FAILURE                                               error
None                    none           10
        LICENSE_FILE_READ_FAILURE                                               error
None                    none           10
        LICENSE_HOSTCONFD_CONNECTION_DOWN                                       error
None                    none           10
        LICENSE_INIT_FAILED                                                     error
None                    none           10
        LICENSE_INTERNAL_INVALID_STRING                                         error
None                    none           10
        LICENSE_INTERNAL_NULL_STRING                                            error
None                    none           10
        LICENSE_INVALID_CONTENT                                                 error
None                    none           10
        LICENSE_INVALID_STRING_IN_FILE                                          error
None                    none           10
        LICENSE_QUOTA_TABLE_CREATE_FAILED                                       error
None                    none           10
        LICENSE_SECONDARY_VALIDATION_COMPROMISED                                error
None                    none           10
        LICENSE_START_DATE_MISSING                                              error
None                    none           10
        LICENSE_STRING_ALLOCATION_FAILURE                                       error
None                    none           10
        LICENSE_TABLE_NOT_PRESENT                                               error
None                    none           10
      operational, Level: error, Plugin: None, Plugin Level: none, Rate limit: 10
     Log ID status:
        LOG ID                                                                  Level
Plugin                  Level          Rate limit
        LICENSE_EXPIRED_INSTALL                                                 error
```

```
None                    none          10
        LICENSE_EXPIRY_CRITICAL                                         error
None                    none          10
        LICENSE_EXPIRY_ERROR                                            error
None                    none          10
        LICENSE_EXPIRY_WARNING                                          error
None                    none          10
        LICENSE_GRACE_PERIOD_ERROR                                      error
None                    none          10
        LICENSE_INSTALL_FAIL                                            error
None                    none          10
        LICENSE_INSTALL_SUCCESSFUL                                      error
None                    none          10
        LICENSE_TRAIL_PERIOD_EXPIRY                                     error
None                    none          10
  lwip_contrib:
    Level: error, Plugin: None, Plugin Level: none
    Log group status:
      generic, Level: error, Plugin: None, Plugin Level: none, Rate limit: 10
     Log ID status:
        LOG ID                                                          Level
Plugin                  Level         Rate limit
        LWIP_AUTH_ENCAP_FAIL                                            error
None                    none          10
        LWIP_AUTH_FAIL                                                  error
None                    none          10
        LWIP_AUTH_FAIL_STAT                                             error
None                    none          10
        LWIP_AUTH_SUCC                                                  error
None                    none          10
        LWIP_CLEAR_AUTH_PCB_STATS                                       error
None                    none          10
        LWIP_CLEAR_AUTH_PCB_STATS_ERR                                   error
None                    none          10
        LWIP_NO_AUTH_KEY                                                error
None                    none          10
        LWIP_TBL_CREATE_FAIL                                            error
None                    none          10
        LWIP_TCP_ACCEPT                                                 error
None                    none          10
        LWIP_TCP_ERROR                                                  error
None                    none          10
  policy:
    Level: error, Plugin: None, Plugin Level: none
    Log group status:
      BDS, Level: error, Plugin: None, Plugin Level: none, Rate limit: 10
     Log ID status:
        LOG ID                                                          Level
Plugin                  Level         Rate limit
        POLICY_LOG_BDS_ERROR                                            error
None                    none          10
```

```
      Configuration, Level: error, Plugin: None, Plugin Level: none, Rate limit: 10
     Log ID status:
        LOG ID                                                                  Level
Plugin                   Level          Rate limit
        POLICY_LOG_CONFIG_PROC_WARNING                                          error
None                     none           10
      Enforcement, Level: error, Plugin: None, Plugin Level: none, Rate limit: 10
      Generation, Level: error, Plugin: None, Plugin Level: none, Rate limit: 10
     Log ID status:
        LOG ID                                                                  Level
Plugin                   Level          Rate limit
        POLICY_LOG_CODE_GEN_ERROR                                               error
None                     none           10
        POLICY_LOG_CODE_GEN_EVENTS                                              error
None                     none           10
        POLICY_LOG_CODE_GEN_NOTIFICATION                                        error
None                     none           10
        POLICY_LOG_CODE_GEN_STATUS_FAILED                                       error
None                     none           10
        POLICY_LOG_CODE_GEN_STATUS_SUCCESS                                      error
None                     none           10
      generic, Level: error, Plugin: None, Plugin Level: none, Rate limit: 10
  pool:
    Level: error, Plugin: None, Plugin Level: none
    Log group status:
      generic, Level: error, Plugin: None, Plugin Level: none, Rate limit: 10
      internal, Level: error, Plugin: None, Plugin Level: none, Rate limit: 10
     Log ID status:
        LOG ID                                                                  Level
Plugin                   Level          Rate limit
        POOL_CLIENT_DICT_ADD_FAILED                                             error
None                     none           10
        POOL_CLIENT_DICT_NOT_INITED                                             error
None                     none           10
        POOL_CLIENT_INIT                                                        error
None                     none           10
        POOL_CLIENT_INVALID_MODULE                                              error
None                     none           10
        POOL_CLIENT_NOT_FOUND                                                   error
None                     none           10
        POOL_CLIENT_OUT_OF_MEMORY                                               error
None                     none           10
        POOL_CLIENT_RANGE_REQ_FAILED                                            error
None                     none           10
        POOL_TABLE_CREATE_FAILED                                                error
None                     none           10
      operational, Level: error, Plugin: None, Plugin Level: none, Rate limit: 10
  pubsub:
    Level: none, Plugin: None, Plugin Level: none
    Log group status:
      generic, Level: none, Plugin: None, Plugin Level: none, Rate limit: 10
```

```
     Log ID status:
        LOG ID                                                          Level
Plugin                    Level          Rate limit
        PUBSUB_INVALID_MESSAGE                                          none
None                      none           10
        PUBSUB_INVALID_PARAMS                                           none
None                      none           10
        PUBSUB_LINE_EVENT                                               none
None                      none           10
        PUBSUB_OBJECT_EVENT                                             none
None                      none           10
        PUBSUB_REGEX_COMPARE_EVENT                                      none
None                      none           10
        PUBSUB_REGEX_EVENT                                              none
None                      none           10
        PUBSUB_REGEX_SCAN_EVENT                                         none
None                      none           10
        PUBSUB_TABLE_EVENT                                              none
None                      none           10
        PUBSUB_TABLE_REGEX_EVENT                                        none
None                      none           10
  secure_management:
    Level: error, Plugin: None, Plugin Level: none
    Log group status:
      general, Level: error, Plugin: None, Plugin Level: none, Rate limit: 10
     Log ID status:
        LOG ID                                                          Level
Plugin                    Level          Rate limit
        SECURE_MANAGEMENT_AUTH_FAIL                                     error
None                      none           10
        SECURE_MANAGEMENT_AUTH_RESOLUTION_ADD_STATS                     error
None                      none           10
        SECURE_MANAGEMENT_AUTH_RESOLUTION_DEL_STATS                     error
None                      none           10
        SECURE_MANAGEMENT_AUTH_SUCCESS                                  error
None                      none           10
        SECURE_MANAGEMENT_AUTH_VLDTN_FAIL                               error
None                      none           10
        SECURE_MANAGEMENT_CREATE_TABLE                                  error
None                      none           10
        SECURE_MANAGEMENT_NO_ATTRIBUTE                                  error
None                      none           10
        SECURE_MANAGEMENT_NO_OBJECT                                     error
None                      none           10
        SECURE_MANAGEMENT_NO_OBJECT_TMPL                                error
None                      none           10
        SECURE_MANAGEMENT_NO_TABLE                                      error
None                      none           10
        SECURE_MANAGEMENT_OBJ_ADD_FAIL                                  error
None                      none           10
        SECURE_MANAGEMENT_OBJ_DEL_FAIL                                  error
```

```
None                    none          10
        SECURE_MANAGEMENT_SBSCR_FAIL                                        error
None                    none          10
      generic, Level: error, Plugin: None, Plugin Level: none, Rate limit: 10
      session-context, Level: info, Plugin: None, Plugin Level: none, Rate limit: 0
     Log ID status:
        LOG ID                                                              Level
Plugin                  Level         Rate limit
        SECURE_MANAGEMENT_USER_AUTH_INVALID_PASSWORD                        info
None                    none          0
        SECURE_MANAGEMENT_USER_AUTH_INVALID_USER                            info
None                    none          0
        SECURE_MANAGEMENT_USER_SESSION_CLOSED                               info
None                    none          0
        SECURE_MANAGEMENT_USER_SESSION_GROUPS                               info
None                    none          0
        SECURE_MANAGEMENT_USER_SESSION_OPENED                               info
None                    none          0
  snapshot:
    Level: error, Plugin: None, Plugin Level: none
    Log group status:
      generic, Level: error, Plugin: None, Plugin Level: none, Rate limit: 10
     Log ID status:
        LOG ID                                                              Level
Plugin                  Level         Rate limit
        SNAPSHOT_TABLE_EVENT                                                error
None                    none          10
  time_series:
    Level: error, Plugin: None, Plugin Level: none
    Log group status:
      generic, Level: error, Plugin: None, Plugin Level: none, Rate limit: 10
     Log ID status:
        LOG ID                                                              Level
Plugin                  Level         Rate limit
        TIME_SERIES_ALERT_INCOMPLETE_CONFIG                                 error
None                    none          10
        TIME_SERIES_ATTRIBUTE_INCOMPLETE_CONFIG                             error
None                    none          10
        TIME_SERIES_ATTRIBUTE_NOT_DEFINED                                   error
None                    none          10
        TIME_SERIES_ATTRIBUTE_TYPE_INVALID                                  error
None                    none          10
        TIME_SERIES_BUFFER_NO_SPACE                                         error
None                    none          10
        TIME_SERIES_CUSTOM_PRINTER_HIT                                      error
None                    none          10
        TIME_SERIES_FILTER_ALLOWED                                          error
None                    none          10
        TIME_SERIES_FILTER_ATTRIBUTE_INCOMPLETE_CONFIG                      error
None                    none          10
        TIME_SERIES_FILTER_DENIED                                           error
```

```
None                    none          10
        TIME_SERIES_FILTER_EVENT                                        error
None                    none          10
        TIME_SERIES_FILTER_LABEL_INCOMPLETE_CONFIG                      error
None                    none          10
        TIME_SERIES_FILTER_MATCH_ATTRIBUTE_TYPE_INVALID                 error
None                    none          10
        TIME_SERIES_FILTER_METRIC_INCOMPLETE_CONFIG                     error
None                    none          10
        TIME_SERIES_INDEX_NAME_INVALID                                  error
None                    none          10
        TIME_SERIES_INVALID_EXTERNAL_LABEL                              error
None                    none          10
        TIME_SERIES_LABEL_ATTRIBUTE_TYPE_INVALID                        error
None                    none          10
        TIME_SERIES_LABEL_INCOMPLETE_CONFIG                             error
None                    none          10
        TIME_SERIES_LABEL_KEY_INVALID_FORMAT                            error
None                    none          10
        TIME_SERIES_LABEL_KEY_SAME_AS_DEFAULT                           error
None                    none          10
        TIME_SERIES_LABEL_NOT_DEFINED                                   error
None                    none          10
        TIME_SERIES_LABEL_TYPE_INVALID                                  error
None                    none          10
        TIME_SERIES_LABEL_VALUE_INVALID                                 error
None                    none          10
        TIME_SERIES_LABEL_VALUE_NOT_DEFINED                             error
None                    none          10
        TIME_SERIES_METRIC_INCOMPLETE_CONFIG                            error
None                    none          10
        TIME_SERIES_METRIC_TYPE_INVALID                                 error
None                    none          10
        TIME_SERIES_NO_ATTRIBUTE                                        error
None                    none          10
        TIME_SERIES_PROCESS_NOT_FOUND                                   error
None                    none          10
        TIME_SERIES_PROMETHEUS_RESTART_EVENT                            error
None                    none          10
        TIME_SERIES_REGEX_EVENT                                         error
None                    none          10
        TIME_SERIES_REG_FAIL                                            error
None                    none          10
        TIME_SERIES_SUBSCRIBED_TABLE                                    error
None                    none          10
        TIME_SERIES_TABLE_NOT_FOUND                                     error
None                    none          10
        TIME_SERIES_TABLE_NOT_FOUND_REG                                 error
None                    none          10
```

#### Viewing BDS Logs

##### Show Log

This command shows BDS logs in the log tables. By default, the command `show log` shows all logs present in the log tables. Various command options are available to filter and display logs. Also, a command option is available to send logs into a file.

**Syntax:**

**show log** <option>

| Option | Description |
|---|---|
| `-` | Without any option, the command displays all logs present in the log tables. |
| `filter <level> <module>` | Render output of the log can be filtered for the specified module or the log level. You can specify filter level or module with any of the `view log` command, so that the logs are filtered based on the specified level or module. |
| `format <abstract \|summary \|detailed>` | Choose any of the three formats: abstract, summary, or detailed. This command provides output in the specified output format. You can specify any of the options at end of any of the `view log` commands to show logs in a specific output format. By default, summary is the log format.<br><br>**Abstract**: Shows logs without metadata<br>**Detailed**: Shows logs in detail with metadata<br>**Summary**: Shows logs in summary view with metadata |
| `table <name>` | Displays logs from a specified log table. Every BD includes multiple log tables. By default, log will be rendered from every log table, if not specified. |
| `to file <filename>` | Name of the file in which logs are transported. You can specify `to file` and the file name at end of any `view log` command to transport the log to the file. |
| `status <bd> <detail> <module>` | Displays log configuration status for the specified BD and module. For more information about log configuration status, see the show log status section. |

Example: Summary view of all logs

```
supervisor@rtbrick: op> show log
[  Notice  ] <2026-06-05T12:52:52.522501+0000> Launch service rtbrick-mribd, PID 309
[  Notice  ] <2026-06-05T12:52:52.537430+0000> Launch service rtbrick-lldpd, PID 307
[  Notice  ] <2026-06-05T12:52:52.560614+0000> Launch service rtbrick-ribd, PID 311
[  Notice  ] <2026-06-05T12:52:52.600276+0000> Launch service rtbrick-staticd, PID 315
```

```
[  Notice  ] <2026-06-05T12:52:52.608738+0000> Launch service rtbrick-ifmd, PID 328
[  Notice  ] <2026-06-05T12:52:52.690550+0000> Launch service rtbrick-fibd, PID 345
[  Notice  ] <2026-06-05T12:56:14.994726+0000> Launch service rtbrick-bgp.appd.1, PID
1027
[  Notice  ] <2026-06-05T12:56:15.052965+0000> Launch service rtbrick-isis.iod.1, PID
1035
[  Notice  ] <2026-06-05T12:56:15.055144+0000> Launch service rtbrick-ospf.appd.1, PID
1038
[  Notice  ] <2026-06-05T12:56:15.059406+0000> Launch service rtbrick-isis.appd.1, PID
1031
[  Notice  ] <2026-06-05T12:56:15.061598+0000> Launch service rtbrick-bgp.iod.1, PID
1029
[  Notice  ] <2026-06-05T12:56:15.063773+0000> Launch service rtbrick-policy.server,
PID 1044
[  Notice  ] <2026-06-05T12:56:15.068176+0000> Launch service rtbrick-ospf.iod.1, PID
1040
[  Notice  ] <2026-06-05T12:59:30.915286+0000> Launch service rtbrick-alertmanager,
PID 1711
[  Error   ] <2026-06-05T12:59:32.586026+0000> Failed to launch service rtbrick-
prometheus
[  Alert   ] <2026-06-05T12:56:34.613858+0000> OAM BFD session is up for peer
2001:db8:1:1::2, local address 2001:db8:1:1::1
[  Alert   ] <2026-06-05T12:56:35.578392+0000> OAM BFD session is up for peer
192.0.2.2, local address 192.0.2.1
[  Alert   ] <2026-06-05T12:56:40.587575+0000> OAM BFD session is up for peer
fe80::e8c5:7aff:fe8f:565f, local address fe80::5c07:58ff:fe79:a17
[  Alert   ] <2026-06-05T12:57:25.575487+0000> OAM BFD session is up for peer
192.0.2.2, local address 192.0.2.1
[  Alert   ] <2026-06-05T12:57:25.575600+0000> OAM BFD session is up for peer
fe80::e8c5:7aff:fe8f:565f, local address fe80::5c07:58ff:fe79:a17
[  Alert   ] <2026-06-05T12:57:28.365348+0000> OAM BFD session is down for peer
fe80::e8c5:7aff:fe8f:565f, local address fe80::5c07:58ff:fe79:a17
[  Alert   ] <2026-06-05T12:57:28.365403+0000> OAM BFD session is down for peer
192.0.2.2, local address 192.0.2.1
[  Alert   ] <2026-06-05T12:57:28.589538+0000> OAM BFD session is up for peer
192.0.2.2, local address 192.0.2.1
[  Alert   ] <2026-06-05T12:57:28.589684+0000> OAM BFD session is up for peer
fe80::e8c5:7aff:fe8f:565f, local address fe80::5c07:58ff:fe79:a17
[  Alert   ] <2026-06-05T12:57:39.876543+0000> OAM BFD session is up for peer
192.0.2.6, local address 192.0.2.5
[  Alert   ] <2026-06-05T12:57:39.876613+0000> OAM BFD session is up for peer
fe80::e8c5:7a00:648f:565f, local address fe80::5c07:5800:6479:a17
[  Alert   ] <2026-06-05T12:58:01.784498+0000> OAM BFD session is up for peer
192.0.2.2, local address 192.0.2.1
[   Info   ] <2026-06-05T13:00:04.090807+0000> User supervisor@10.200.137.164
(session-id 4) executed command 'commit' via cli non-interactive
[   Info   ] <2026-06-05T13:00:05.420634+0000> User supervisor@10.200.137.164
(session-id 4) executed command 'show log status' via cli non-interactive
[   Info   ] <2026-06-05T13:00:06.931263+0000> User supervisor@10.200.137.164
(session-id 4) executed command 'show log status module bgp' via cli non-interactive
[   Info   ] <2026-06-05T13:00:08.034992+0000> User supervisor@10.200.137.164
```

```
(session-id 4) executed command 'show log status bd bgp.appd.1' via cli non-
interactive
[   Info   ] <2026-06-05T13:00:09.386707+0000> User supervisor@10.200.137.164
(session-id 4) executed command 'show log status module bgp bd bgp.appd.1' via cli
non-interactive
[   Info   ] <2026-06-05T13:00:10.542072+0000> User supervisor@10.200.137.164
(session-id 4) executed command 'show log status bd bgp.appd.1 detail' via cli non-
interactive
[  Alert   ] <2026-06-05T12:56:25.343430+0000> ISIS instance default, level 1, exit,
overload state
[  Alert   ] <2026-06-05T12:56:25.344120+0000> ISIS instance default, level 2, exit,
overload state
[  Alert   ] <2026-06-05T12:56:33.603720+0000> ISIS neighbor 1980.5110.0121.00,
discovered, interface ifl-0/1/22/0, instance default
[  Alert   ] <2026-06-05T12:56:44.336313+0000> ISIS neighbor 1980.5110.0121.00,
interface ifl-0/1/22/0, instance default, changed state from Down, to Up, reason
[  Alert   ] <2026-06-05T12:58:05.859673+0000> ISIS neighbor 1980.5110.0121.00,
interface ifl-0/1/22/0, instance default, changed state from Up, to Down, reason Admin
reset
[  Alert   ] <2026-06-05T12:56:34.463596+0000> BGP FSM change, peer 2001:db8:1:1::2,
source 2001:db8:1:1::1, hostname PE2, instance default changed state from Idle to
Established, reason None
[  Alert   ] <2026-06-05T12:56:35.434680+0000> BGP FSM change, peer 192.0.2.2, source
192.0.2.1, hostname PE2, instance default changed state from Idle to Established,
reason None
[  Alert   ] <2026-06-05T12:56:40.436244+0000> BGP FSM change, peer
fe80::e8c5:7aff:fe8f:565f, source fe80::5c07:58ff:fe79:a17, hostname PE2, instance
default changed state from Idle to Established, reason None
[  Alert   ] <2026-06-05T12:58:05.855520+0000> BGP FSM change, peer
fe80::e8c5:7aff:fe8f:565f, source fe80::5c07:58ff:fe79:a17, hostname -, instance
default changed state from Established to Idle, reason Cease, Sub-Code: Admin shutdown
[  Alert   ] <2026-06-05T12:58:05.856055+0000> BGP FSM change, peer 192.0.2.2, source
192.0.2.1, hostname -, instance default changed state from Established to Idle, reason
Cease, Sub-Code: Admin shutdown
[  Alert   ] <2026-06-05T12:58:05.856654+0000> BGP FSM change, peer 2001:db8:1:1::2,
source 2001:db8:1:1::1, hostname -, instance default changed state from Established to
Idle, reason Cease, Sub-Code: Admin shutdown
```

Example: Summary view for the specified log table

```
supervisor@rtbrick: op> show log table secure_management.confd.log
[   Info   ] <2026-06-05T13:00:12.254878+0000> Session closed for user supervisor from
10.200.137.164 with session-id 4
[   Info   ] <2026-06-05T13:00:12.385841+0000> Authentication successful for user
supervisor from 10.200.137.164 with session-id 12
[   Info   ] <2026-06-05T13:00:12.883866+0000> Group memberships for user supervisor
with session-id 12: supervisor debug_info rtbrick_confd_sock_access
```

Example: Summary view of specified filters on all logs from a single table

```
supervisor@rtbrick: op> show log table cli-cmd.confd.log filter level info
[   Info   ] <2026-06-05T13:00:04.090807+0000> User supervisor@10.200.137.164
(session-id 4) executed command 'commit' via cli non-interactive
[   Info   ] <2026-06-05T13:00:05.420634+0000> User supervisor@10.200.137.164
(session-id 4) executed command 'show log status' via cli non-interactive
[   Info   ] <2026-06-05T13:00:06.931263+0000> User supervisor@10.200.137.164
(session-id 4) executed command 'show log status module bgp' via cli non-interactive
[   Info   ] <2026-06-05T13:00:08.034992+0000> User supervisor@10.200.137.164
(session-id 4) executed command 'show log status bd bgp.appd.1' via cli non-
interactive
[   Info   ] <2026-06-05T13:00:09.386707+0000> User supervisor@10.200.137.164
(session-id 4) executed command 'show log status module bgp bd bgp.appd.1' via cli
non-interactive
[   Info   ] <2026-06-05T13:00:10.542072+0000> User supervisor@10.200.137.164
(session-id 4) executed command 'show log status bd bgp.appd.1 detail' via cli non-
interactive
[   Info   ] <2026-06-05T13:00:11.737004+0000> User supervisor@10.200.137.164
(session-id 4) executed command 'show log' via cli non-interactive
[   Info   ] <2026-06-05T13:00:14.395389+0000> User supervisor@10.200.137.164
(session-id 12) executed command 'show log table secure_management.confd.log' via cli
non-interactive
[   Info   ] <2026-06-05T13:00:15.620891+0000> User supervisor@10.200.137.164
(session-id 12) executed command 'set system user user1 role supervisor' via cli non-
interactive
[   Info   ] <2026-06-05T13:00:16.400793+0000> User supervisor@10.200.137.164
(session-id 12) executed command 'discard' via cli non-interactive
```

Example: Exporting logs to a file

```
supervisor@rtbrick:~ $ cat test.log
[  notice  ] <2026-06-05T12:52:52.522501+0000> Launch service rtbrick-mribd, PID 309
[  notice  ] <2026-06-05T12:52:52.537430+0000> Launch service rtbrick-lldpd, PID 307
[  notice  ] <2026-06-05T12:52:52.560614+0000> Launch service rtbrick-ribd, PID 311
[  notice  ] <2026-06-05T12:52:52.600276+0000> Launch service rtbrick-staticd, PID 315
[  notice  ] <2026-06-05T12:52:52.608738+0000> Launch service rtbrick-ifmd, PID 328
[  notice  ] <2026-06-05T12:52:52.690550+0000> Launch service rtbrick-fibd, PID 345
[  notice  ] <2026-06-05T12:56:14.994726+0000> Launch service rtbrick-bgp.appd.1, PID
1027
[  notice  ] <2026-06-05T12:56:15.052965+0000> Launch service rtbrick-isis.iod.1, PID
1035
[  notice  ] <2026-06-05T12:56:15.055144+0000> Launch service rtbrick-ospf.appd.1, PID
1038
[  notice  ] <2026-06-05T12:56:15.059406+0000> Launch service rtbrick-isis.appd.1, PID
1031
[  notice  ] <2026-06-05T12:56:15.061598+0000> Launch service rtbrick-bgp.iod.1, PID
1029
[  notice  ] <2026-06-05T12:56:15.063773+0000> Launch service rtbrick-policy.server,
PID 1044
[  notice  ] <2026-06-05T12:56:15.068176+0000> Launch service rtbrick-ospf.iod.1, PID
```

```
1040
[  notice  ] <2026-06-05T12:59:30.915286+0000> Launch service rtbrick-alertmanager,
PID 1711
[  error   ] <2026-06-05T12:59:32.586026+0000> Failed to launch service rtbrick-
prometheus
[  alert   ] <2026-06-05T12:56:34.613858+0000> OAM BFD session is up for peer
2001:db8:1:1::2, local address 2001:db8:1:1::1
[  alert   ] <2026-06-05T12:56:35.578392+0000> OAM BFD session is up for peer
192.0.2.2, local address 192.0.2.1
[  alert   ] <2026-06-05T12:56:40.587575+0000> OAM BFD session is up for peer
fe80::e8c5:7aff:fe8f:565f, local address fe80::5c07:58ff:fe79:a17
[  alert   ] <2026-06-05T12:57:25.575487+0000> OAM BFD session is up for peer
192.0.2.2, local address 192.0.2.1
[  alert   ] <2026-06-05T12:57:25.575600+0000> OAM BFD session is up for peer
fe80::e8c5:7aff:fe8f:565f, local address fe80::5c07:58ff:fe79:a17
[  alert   ] <2026-06-05T12:57:28.365348+0000> OAM BFD session is down for peer
fe80::e8c5:7aff:fe8f:565f, local address fe80::5c07:58ff:fe79:a17
[  alert   ] <2026-06-05T12:57:28.365403+0000> OAM BFD session is down for peer
192.0.2.2, local address 192.0.2.1
[  alert   ] <2026-06-05T12:57:28.589538+0000> OAM BFD session is up for peer
192.0.2.2, local address 192.0.2.1
[  alert   ] <2026-06-05T12:57:28.589684+0000> OAM BFD session is up for peer
fe80::e8c5:7aff:fe8f:565f, local address fe80::5c07:58ff:fe79:a17
[  alert   ] <2026-06-05T12:57:39.876543+0000> OAM BFD session is up for peer
192.0.2.6, local address 192.0.2.5
[  alert   ] <2026-06-05T12:57:39.876613+0000> OAM BFD session is up for peer
fe80::e8c5:7a00:648f:565f, local address fe80::5c07:5800:6479:a17
[  alert   ] <2026-06-05T12:58:01.784498+0000> OAM BFD session is up for peer
192.0.2.2, local address 192.0.2.1
[   info   ] <2026-06-05T13:00:04.090807+0000> User supervisor@10.200.137.164
(session-id 4) executed command 'commit' via cli non-interactive
[   info   ] <2026-06-05T13:00:05.420634+0000> User supervisor@10.200.137.164
(session-id 4) executed command 'show log status' via cli non-interactive
[   info   ] <2026-06-05T13:00:06.931263+0000> User supervisor@10.200.137.164
(session-id 4) executed command 'show log status module bgp' via cli non-interactive
[   info   ] <2026-06-05T13:00:08.034992+0000> User supervisor@10.200.137.164
(session-id 4) executed command 'show log status bd bgp.appd.1' via cli non-
interactive
[   info   ] <2026-06-05T13:00:09.386707+0000> User supervisor@10.200.137.164
(session-id 4) executed command 'show log status module bgp bd bgp.appd.1' via cli
non-interactive
[   info   ] <2026-06-05T13:00:10.542072+0000> User supervisor@10.200.137.164
(session-id 4) executed command 'show log status bd bgp.appd.1 detail' via cli non-
interactive
[   info   ] <2026-06-05T13:00:11.737004+0000> User supervisor@10.200.137.164
(session-id 4) executed command 'show log' via cli non-interactive
[   info   ] <2026-06-05T13:00:14.395389+0000> User supervisor@10.200.137.164
(session-id 12) executed command 'show log table secure_management.confd.log' via cli
non-interactive
[   info   ] <2026-06-05T13:00:15.620891+0000> User supervisor@10.200.137.164
(session-id 12) executed command 'set system user user1 role supervisor' via cli non-
```

```
interactive
[   info   ] <2026-06-05T13:00:16.400793+0000> User supervisor@10.200.137.164
(session-id 12) executed command 'discard' via cli non-interactive
[   info   ] <2026-06-05T13:00:17.389973+0000> User supervisor@10.200.137.164
(session-id 12) executed command 'show log table cli-cmd.confd.log filter level info'
via cli non-interactive
[  alert   ] <2026-06-05T12:56:25.343430+0000> ISIS instance default, level 1, exit,
overload state
[  alert   ] <2026-06-05T12:56:25.344120+0000> ISIS instance default, level 2, exit,
overload state
[  alert   ] <2026-06-05T12:56:33.603720+0000> ISIS neighbor 1980.5110.0121.00,
discovered, interface ifl-0/1/22/0, instance default
[  alert   ] <2026-06-05T12:56:44.336313+0000> ISIS neighbor 1980.5110.0121.00,
interface ifl-0/1/22/0, instance default, changed state from Down, to Up, reason
[  alert   ] <2026-06-05T12:58:05.859673+0000> ISIS neighbor 1980.5110.0121.00,
interface ifl-0/1/22/0, instance default, changed state from Up, to Down, reason Admin
reset
[  alert   ] <2026-06-05T12:56:34.463596+0000> BGP FSM change, peer 2001:db8:1:1::2,
source 2001:db8:1:1::1, hostname PE2, instance default changed state from Idle to
Established, reason None
[  alert   ] <2026-06-05T12:56:35.434680+0000> BGP FSM change, peer 192.0.2.2, source
192.0.2.1, hostname PE2, instance default changed state from Idle to Established,
reason None
[  alert   ] <2026-06-05T12:56:40.436244+0000> BGP FSM change, peer
fe80::e8c5:7aff:fe8f:565f, source fe80::5c07:58ff:fe79:a17, hostname PE2, instance
default changed state from Idle to Established, reason None
[  alert   ] <2026-06-05T12:58:05.855520+0000> BGP FSM change, peer
fe80::e8c5:7aff:fe8f:565f, source fe80::5c07:58ff:fe79:a17, hostname -, instance
default changed state from Established to Idle, reason Cease, Sub-Code: Admin shutdown
[  alert   ] <2026-06-05T12:58:05.856055+0000> BGP FSM change, peer 192.0.2.2, source
192.0.2.1, hostname -, instance default changed state from Established to Idle, reason
Cease, Sub-Code: Admin shutdown
[  alert   ] <2026-06-05T12:58:05.856654+0000> BGP FSM change, peer 2001:db8:1:1::2,
source 2001:db8:1:1::1, hostname -, instance default changed state from Established to
Idle, reason Cease, Sub-Code: Admin shutdown
[   info   ] <2026-06-05T13:00:12.254878+0000> Session closed for user supervisor from
10.200.137.164 with session-id 4
[   info   ] <2026-06-05T13:00:12.385841+0000> Authentication successful for user
supervisor from 10.200.137.164 with session-id 12
[   info   ] <2026-06-05T13:00:12.883866+0000> Group memberships for user supervisor
with session-id 12: supervisor debug_info rtbrick_confd_sock_access
```

Example: Summary of the logs in an abstract format

```
supervisor@rtbrick: op> show log format abstract
Launch service rtbrick-mribd, PID 309
Launch service rtbrick-lldpd, PID 307
Launch service rtbrick-ribd, PID 311
Launch service rtbrick-staticd, PID 315
Launch service rtbrick-ifmd, PID 328
```

```
Launch service rtbrick-fibd, PID 345
Launch service rtbrick-bgp.appd.1, PID 1027
Launch service rtbrick-isis.iod.1, PID 1035
Launch service rtbrick-ospf.appd.1, PID 1038
Launch service rtbrick-isis.appd.1, PID 1031
Launch service rtbrick-bgp.iod.1, PID 1029
Launch service rtbrick-policy.server, PID 1044
Launch service rtbrick-ospf.iod.1, PID 1040
Launch service rtbrick-alertmanager, PID 1711
Failed to launch service rtbrick-prometheus
OAM BFD session is up for peer 2001:db8:1:1::2, local address 2001:db8:1:1::1
OAM BFD session is up for peer 192.0.2.2, local address 192.0.2.1
OAM BFD session is up for peer fe80::e8c5:7aff:fe8f:565f, local address
fe80::5c07:58ff:fe79:a17
OAM BFD session is up for peer 192.0.2.2, local address 192.0.2.1
OAM BFD session is up for peer fe80::e8c5:7aff:fe8f:565f, local address
fe80::5c07:58ff:fe79:a17
OAM BFD session is down for peer fe80::e8c5:7aff:fe8f:565f, local address
fe80::5c07:58ff:fe79:a17
OAM BFD session is down for peer 192.0.2.2, local address 192.0.2.1
OAM BFD session is up for peer 192.0.2.2, local address 192.0.2.1
OAM BFD session is up for peer fe80::e8c5:7aff:fe8f:565f, local address
fe80::5c07:58ff:fe79:a17
OAM BFD session is up for peer 192.0.2.6, local address 192.0.2.5
OAM BFD session is up for peer fe80::e8c5:7a00:648f:565f, local address
fe80::5c07:5800:6479:a17
OAM BFD session is up for peer 192.0.2.2, local address 192.0.2.1
User supervisor@10.200.137.164 (session-id 4) executed command 'commit' via cli non-
interactive
User supervisor@10.200.137.164 (session-id 4) executed command 'show log status' via
cli non-interactive
User supervisor@10.200.137.164 (session-id 4) executed command 'show log status module
bgp' via cli non-interactive
User supervisor@10.200.137.164 (session-id 4) executed command 'show log status bd
bgp.appd.1' via cli non-interactive
User supervisor@10.200.137.164 (session-id 4) executed command 'show log status module
bgp bd bgp.appd.1' via cli non-interactive
User supervisor@10.200.137.164 (session-id 4) executed command 'show log status bd
bgp.appd.1 detail' via cli non-interactive
User supervisor@10.200.137.164 (session-id 4) executed command 'show log' via cli non-
interactive
User supervisor@10.200.137.164 (session-id 12) executed command 'show log table
secure_management.confd.log' via cli non-interactive
User supervisor@10.200.137.164 (session-id 12) executed command 'set system user user1
role supervisor' via cli non-interactive
User supervisor@10.200.137.164 (session-id 12) executed command 'discard' via cli non-
interactive
User supervisor@10.200.137.164 (session-id 12) executed command 'show log table cli-
cmd.confd.log filter level info' via cli non-interactive
User supervisor@10.200.137.164 (session-id 12) executed command 'show log to file
test.log' via cli non-interactive
```

```
User supervisor@10.200.137.164 (session-id 12) executed command 'show log filter level
error' via cli non-interactive
User supervisor@10.200.137.164 (session-id 12) executed command 'show log filter
module secure_management' via cli non-interactive
ISIS instance default, level 1, exit, overload state
ISIS instance default, level 2, exit, overload state
ISIS neighbor 1980.5110.0121.00, discovered, interface ifl-0/1/22/0, instance default
ISIS neighbor 1980.5110.0121.00, interface ifl-0/1/22/0, instance default, changed
state from Down, to Up, reason
ISIS neighbor 1980.5110.0121.00, interface ifl-0/1/22/0, instance default, changed
state from Up, to Down, reason Admin reset
BGP FSM change, peer 2001:db8:1:1::2, source 2001:db8:1:1::1, hostname PE2, instance
default changed state from Idle to Established, reason None
BGP FSM change, peer 192.0.2.2, source 192.0.2.1, hostname PE2, instance default
changed state from Idle to Established, reason None
BGP FSM change, peer fe80::e8c5:7aff:fe8f:565f, source fe80::5c07:58ff:fe79:a17,
hostname PE2, instance default changed state from Idle to Established, reason None
BGP FSM change, peer fe80::e8c5:7aff:fe8f:565f, source fe80::5c07:58ff:fe79:a17,
hostname -, instance default changed state from Established to Idle, reason Cease,
Sub-Code: Admin shutdown
BGP FSM change, peer 192.0.2.2, source 192.0.2.1, hostname -, instance default changed
state from Established to Idle, reason Cease, Sub-Code: Admin shutdown
BGP FSM change, peer 2001:db8:1:1::2, source 2001:db8:1:1::1, hostname -, instance
default changed state from Established to Idle, reason Cease, Sub-Code: Admin shutdown
Session closed for user supervisor from 10.200.137.164 with session-id 4
Authentication successful for user supervisor from 10.200.137.164 with session-id 12
Group memberships for user supervisor with session-id 12: supervisor debug_info
rtbrick_confd_sock_access
```

#### BDS Logging Clear Commands

Clear commands allow you to delete existing logs.

#### Clear Logs

This commands resets all logs.

**Syntax:**

**clear log** <option> <attribute> <value>

| Attribute | Description |
|---|---|
| `bd <bd-name>` | Clear all BDS logs from the given BD. |
| `table <table-name>` | Clears the specified log table. |

### Viewing RtBrick Host Log Files

RtBrick Host log files are available at the following directory:

CtrlD log files are available at:

`/var/log/rtbrick-ctrld.log`

Example: CtrlD Logs

```
supervisor@host>rtbrick:~ $ tail -10 /var/log/rtbrick-ctrld-service-err.log
time=2026-06-05T13:00:19.146Z level=INFO msg="HTTP request completed"
duration=23.449848ms remote_addr=127.0.0.1:49544 host=localhost:19091 method=GET
path=/api/v1/ctrld/elements request_id=b8530f2c-9413-4c16-ba67-905ea4546a2c
statuscode=200 user_name="" user_subject=""
time=2026-06-05T13:00:21.179Z level=INFO msg="HTTP request completed"
duration=28.072562ms remote_addr=127.0.0.1:49544 host=localhost:19091 method=GET
path=/api/v1/ctrld/elements request_id=fe28a4ab-02a5-46cb-b177-61ad3fee4e85
statuscode=200 user_name="" user_subject=""
time=2026-06-05T13:00:23.211Z level=INFO msg="HTTP request completed"
duration=22.873176ms remote_addr=127.0.0.1:49544 host=localhost:19091 method=GET
path=/api/v1/ctrld/elements request_id=6987cb43-0bd6-4bab-ae3f-406b064dc1d6
statuscode=200 user_name="" user_subject=""
time=2026-06-05T13:00:23.663Z level=INFO msg="HTTP request completed"
duration=403.514µs remote_addr=10.200.129.252:35362
host=ufi17.q2c.u34.r6.nbg.rtbrick.net:19091 method=GET path=/metrics
request_id=c83744c5-539d-402a-b172-1a5a5c2a8eb6 statuscode=0 user_name=""
user_subject=""
time=2026-06-05T13:00:23.695Z level=INFO msg="HTTP request completed"
duration=1.372579ms remote_addr=10.200.129.252:35378
host=ufi17.q2c.u34.r6.nbg.rtbrick.net:19091 method=GET
path=/api/v1/rbfs/elements/ufi17.q2c.u34.r6.nbg.rtbrick.net/services/prometheus/proxy/
federate request_id=0eb35687-0279-4cdf-830d-121cbe3ffb4a statuscode=200 user_name=""
user_subject=""
time=2026-06-05T13:00:25.156Z level=INFO msg="HTTP request completed"
duration=23.511437ms remote_addr=127.0.0.1:49544 host=localhost:19091 method=GET
path=/api/v1/ctrld/elements request_id=dc38cd63-c329-4cf9-be92-3f31f45dd3e1
statuscode=200 user_name="" user_subject=""
time=2026-06-05T13:00:27.188Z level=INFO msg="HTTP request completed"
duration=26.737784ms remote_addr=127.0.0.1:49544 host=localhost:19091 method=GET
path=/api/v1/ctrld/elements request_id=89f53399-dcc2-43cb-b07a-d76633b63f17
statuscode=200 user_name="" user_subject=""
time=2026-06-05T13:00:28.663Z level=INFO msg="HTTP request completed"
duration=385.626µs remote_addr=10.200.129.252:35362
host=ufi17.q2c.u34.r6.nbg.rtbrick.net:19091 method=GET path=/metrics
request_id=ed5f1dde-a162-4657-a217-64342b26b6b3 statuscode=0 user_name=""
user_subject=""
time=2026-06-05T13:00:28.695Z level=INFO msg="HTTP request completed"
duration=1.417989ms remote_addr=10.200.129.252:35378
host=ufi17.q2c.u34.r6.nbg.rtbrick.net:19091 method=GET
path=/api/v1/rbfs/elements/ufi17.q2c.u34.r6.nbg.rtbrick.net/services/prometheus/proxy/
federate request_id=a0ac4183-2af6-469a-ac3d-0f321f0bf847 statuscode=200 user_name=""
user_subject=""
time=2026-06-05T13:00:29.207Z level=INFO msg="HTTP request completed"
duration=23.591276ms remote_addr=127.0.0.1:49544 host=localhost:19091 method=GET
```

```
path=/api/v1/ctrld/elements request_id=41e0f5dd-8eef-43d6-abd4-7291ab4dfe57
statuscode=200 user_name="" user_subject=""
```

ApiGwD log files are available at:

`/var/log/rtbrick-apigwd.log`

Example: API Gateway Logs

```
supervisor@host>rtbrick:~ $ cat /var/log/rtbrick-apigwd-service-err.log
time=2026-06-05T12:52:33.570Z level=INFO msg="watching for file change"
file=/etc/rtbrick/apigwd/config.json
time=2026-06-05T12:52:33.576Z level=INFO msg="Added ip address to the key"
ip=127.0.0.1
time=2026-06-05T12:52:33.576Z level=INFO msg="Added ip address to the key" ip=10.0.3.1
time=2026-06-05T12:52:33.576Z level=INFO msg="Added ip address to the key" ip=::1
time=2026-06-05T12:52:34.007Z level=INFO msg="Generated new TLS key"
file=/etc/rtbrick/apigwd/tls.pem
time=2026-06-05T12:52:34.010Z level=INFO msg="certificate and key loaded"
time=2026-06-05T12:52:34.010Z level=INFO msg="watching for cert and key change"
time=2026-06-05T12:52:34.010Z level=INFO msg="Serve APIGWD Server" addr=[::]:12321
time=2026-06-05T12:53:23.655Z level=INFO msg="watch event"
event="{Name:/etc/rtbrick/apigwd/config.json Op:2}"
time=2026-06-05T12:53:23.655Z level=INFO msg="reloaded request limiter config"
time=2026-06-05T12:53:23.657Z level=INFO msg="reloaded access config"
time=2026-06-05T12:53:23.658Z level=INFO msg="watch event: WRITE
\"/etc/rtbrick/apigwd/tls.pem\""
time=2026-06-05T12:53:23.660Z level=INFO msg="certificate and key loaded"
time=2026-06-05T12:53:23.660Z level=INFO msg="watch event: WRITE
\"/etc/rtbrick/apigwd/tls.pem\""
time=2026-06-05T12:53:23.661Z level=INFO msg="certificate and key loaded"
```

**Viewing Logs in Graylog**

For viewing your log data on Graylog, perform the following steps:

1. Open the Graylog webpage.
2. Log in using your user credentials.

*Figure 169. Log Into Graylog*

3. Click **System** and select **Input**.

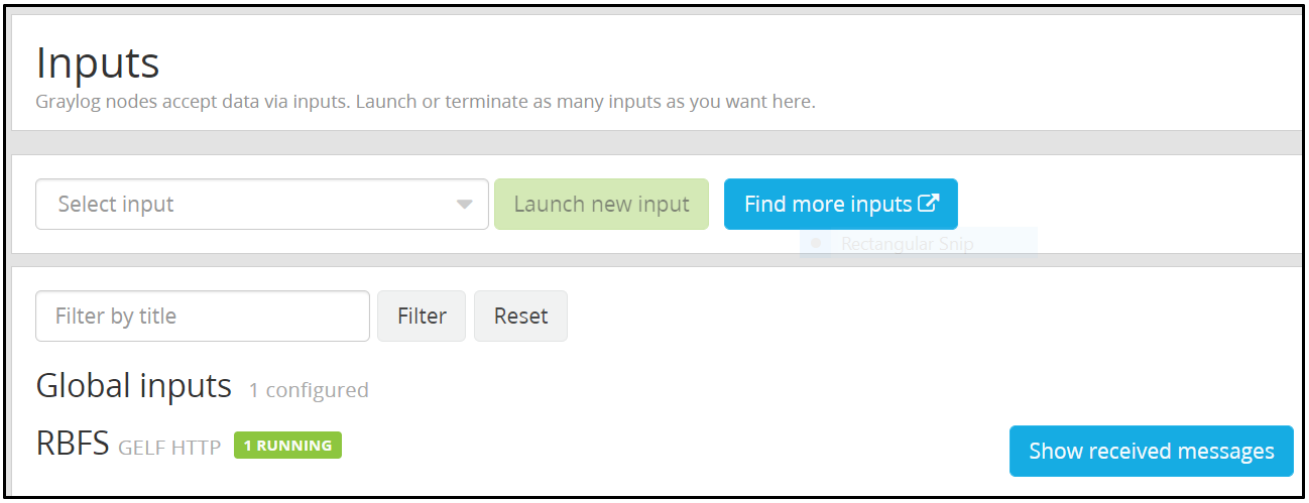



*Figure 170. Graylog Inputs*

4. Click the **Show received message** tab.

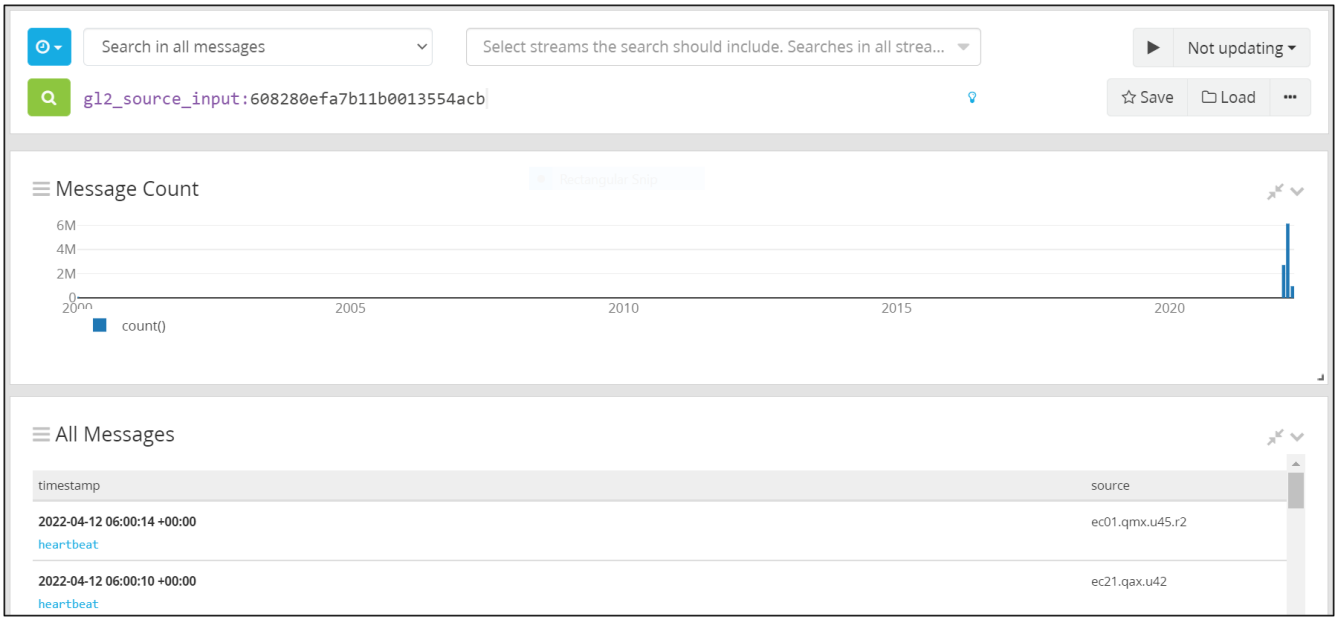



*Figure 171. Viewing Logs in Graylog*

The log messages page appears.

## 11.3.4. RBFS Log Reference

RBFS generates logs for various events. RBFS logs are record of events which occur on different functional components or modules in RBFS.

RBFS logs are categorized based on demons or modules from which they are produced. The following are some of the example scenarios when event logs are generated:

- Failure conditions such as failure in decoding received DHCP relay packet or failure to access a configuration file.
- Normal operations such as when an IPoE subscriber is created or deleted and an interface has been configured and so on.

- Critical conditions such as power failure.

This document provides log information such as log ID, message, description and module, severity level and recommended actions.

Log Destination You can locate and view the logs using multiple interfaces which include RBFS command line interface. RBFS also supports exporting logs to external log management servers such as Syslog server and Graylog server for real-time analysis.

You can view the logs in the RBFS command line interface using the `show log` command. Various command options are available to filter and display logs. For more information, see the Logging Operational Commands section.

### bd Log Messages

#### Log Level Summary

Error(20), Notice(2), Info(5)

| | |
|---|---|
| **Log ID** | BD_UV_UDP_BIND_FAILED |
| **Message** | UV_UDP: UDP socket bind failed for addr: %s port: %s with error: %s |
| **Description** | Failed to bind to local addr and port to create UDP socket |
| **Log Module** | bd |
| **Log Group** | udp |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BD_UV_UDP_CONNECT_FAILED |
| **Message** | UV_UDP: UDP socket connect failed for remote addr: %s port: %s with error: %s |
| **Description** | Failed to connect to remote addr and port to create UDP socket |
| **Log Module** | bd |
| **Log Group** | udp |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BD_UV_UDP_INVALID_ADDR_FAMILY |
| **Message** | UV_UDP: UDP socket creation failed due to invalid address family %s |

| | |
|---|---|
| **Description** | Failed to create UDP socket due to invalid address family |
| **Log Module** | bd |
| **Log Group** | udp |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BD_UV_UDP_READ_START_FAILED |
| **Message** | UV_UDP: Failed to start read on UDP socket add: %s port: %s with error: %s |
| **Description** | Failed to start read on UDP socket |
| **Log Module** | bd |
| **Log Group** | udp |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BD_UV_UDP_MISSING_CB |
| **Message** | UV_UDP: UDP socket create failed due to missing receive or send callback |
| **Description** | UDP socket create failed due to missing receive or send callback |
| **Log Module** | bd |
| **Log Group** | udp |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BD_UV_DNS_TEMPLATE_MISSING_CB |
| **Message** | UV_DNS: DNS getnameinfo failed due to missing paramters |
| **Description** | DNS getnameinfo failed due to missing paramters |
| **Log Module** | bd |
| **Log Group** | dns-resolver |
| **Severity** | ERROR |

| | |
|---|---|
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BD_UV_GETNAMEINFO_FAILED |
| **Message** | UV_DNS: DNS getnameinfo failed to fetch due to %s |
| **Description** | DNS getnameinfo failed to fetch |
| **Log Module** | bd |
| **Log Group** | dns-resolver |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BD_SERVICE_START_FAILED |
| **Message** | Failed to launch service %s |
| **Description** | Failed launch new service |
| **Log Module** | bd |
| **Log Group** | service-launch |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BD_SERVICE_STOP_FAILED |
| **Message** | Failed to terminate service %s |
| **Description** | Terminate service |
| **Log Module** | bd |
| **Log Group** | service-launch |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BD_SERVICE_RESTART_FAILED |
| **Message** | Failed to restart service %s |

| | |
|---|---|
| **Description** | Failed to restart service |
| **Log Module** | bd |
| **Log Group** | service-launch |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | HTTP_JWK_FILE_READ_FAILED |
| **Message** | Failed to read keys from JWK file %s |
| **Description** | Failed to read keys from JWK file |
| **Log Module** | bd |
| **Log Group** | http |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | HTTP_JWK_MULTIPLE_KEYS |
| **Message** | Multiple keys in JWK secret file |
| **Description** | Multiple keys in JWK secret file |
| **Log Module** | bd |
| **Log Group** | http |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | HTTP_JWK_FILE_MISSING |
| **Message** | JSON Web Key file %s not found |
| **Description** | JSON Web Key file not found |
| **Log Module** | bd |
| **Log Group** | http |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | HTTP_JWK_FILE_MEM_ALLOC_FAILED |
| **Message** | Memory error: unable to allocate %s bytes |
| **Description** | Unable to allocate memory to read JSON Wen key file |
| **Log Module** | bd |
| **Log Group** | http |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | HTTP_JWK_FILE_OPEN_FAILED |
| **Message** | Unable to open JWK file %s |
| **Description** | Unable to open JWK file |
| **Log Module** | bd |
| **Log Group** | http |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | HTTP_JWK_MISSING_KEY |
| **Message** | Key missing in JWK secret file |
| **Description** | Key missing in JWK secret file |
| **Log Module** | bd |
| **Log Group** | http |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | HTTP_JWK_FILE_JSON_PARSE_FAILED |
| **Message** | Failed to parse json content from JWK file %s |
| **Description** | Failed to parse json content from JWK file |
| **Log Module** | bd |

| | |
|---|---|
| **Log Group** | http |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | HTTP_WRITE_BUFFER_MEM_ALLOC_FAILED |
| **Message** | Failed to allocate memory for HTTP write buffer of size %s |
| **Description** | Failed to allocate memory for HTTP write buffer |
| **Log Module** | bd |
| **Log Group** | http |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | HTTP_WRITE_PRINTF_FAILED |
| **Message** | Failed to write data to http write buffer |
| **Description** | Failed to write data to http write buffer |
| **Log Module** | bd |
| **Log Group** | http |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | HTTP_RESP_INSUFFICIENT_BUFFER |
| **Message** | Failed to read response body, body len: %s input buffer len: %s |
| **Description** | Failed to read response body due to insufficient buffer |
| **Log Module** | bd |
| **Log Group** | http |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BD_SERVICE_START |
| **Message** | Launch service %s, PID %s |
| **Description** | Launch new service |
| **Log Module** | bd |
| **Log Group** | service-launch |
| **Severity** | NOTICE |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BD_SERVICE_STOP |
| **Message** | Terminate service %s |
| **Description** | Terminate service |
| **Log Module** | bd |
| **Log Group** | service-launch |
| **Severity** | NOTICE |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BD_SERVICE_START_REQUEST |
| **Message** | %s module requested to launch %s service |
| **Description** | Service launch request from module |
| **Log Module** | bd |
| **Log Group** | service-launch |
| **Severity** | INFO |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BD_SERVICE_STOP_REQUEST |
| **Message** | %s module requested to terminate service %s |
| **Description** | Service terminate request from module |
| **Log Module** | bd |
| **Log Group** | service-launch |
| **Severity** | INFO |

| | |
|---|---|
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BD_SERVICE_RESTART |
| **Message** | Restart service %s, PID %s |
| **Description** | Restart new service |
| **Log Module** | bd |
| **Log Group** | service-launch |
| **Severity** | INFO |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | HTTP_SEND |
| **Message** | Send response for request endpoint %s %s buf count %s state %s |
| **Description** | Send response details |
| **Log Module** | bd |
| **Log Group** | http |
| **Severity** | INFO |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | HTTP_CLIENT_REQUEST_LOG |
| **Message** | method: %s path: %s request body: %s response code: %s response body: %s |
| **Description** | Log http request and response |
| **Log Module** | bd |
| **Log Group** | http |
| **Severity** | INFO |
| **Recommended Actions** | Contact Rtbrick customer support |

## BGP Log Messages

### Log Level Summary

Alert(1), Notice(1), Info(45), Debug(16)

| | |
|---|---|
| **Log ID** | BGP_PEER_STATE_CHANGE |
| **Message** | BGP FSM change, peer %s, source %s, hostname %s, instance %s changed state from %s to %s, reason %s |
| **Description** | BGP FSM state change log |
| **Log Module** | bgp |
| **Log Group** | peer |
| **Severity** | Alert |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_PEER_RESET |
| **Message** | BGP peer %s, source %s, hostname %s, instance %s reset, reason %s |
| **Description** | BGP Session reset |
| **Log Module** | bgp |
| **Log Group** | peer |
| **Severity** | Notice |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_IFL_NBR_ADD |
| **Message** | Link-local peer %s discovered on interface %s, instance %s |
| **Description** | BGP Link-local peer discovered on interface |
| **Log Module** | bgp |
| **Log Group** | peer |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_IFL_NBR_DEL |

| | |
|---|---|
| **Message** | Link-local peer %s deleted on interface %s, instance %s |
| **Description** | BGP Link-local peer deleted on interface |
| **Log Module** | bgp |
| **Log Group** | peer |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_PEER_RESET_FSM_ERR |
| **Message** | BGP peer %s, source %s, hostname %s, instance %s reset, reason FSM error subcode %s, message type %s |
| **Description** | BGP Session reset due to FSM error |
| **Log Module** | bgp |
| **Log Group** | peer |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_MSG_OPEN_VERSION |
| **Message** | BGP peer %s, source %s, hostname %s, instance %s received open message with version %s |
| **Description** | BGP peer %s, source %s, hostname %s, instance %s received open message with unsupported version |
| **Log Module** | bgp |
| **Log Group** | peer |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_MSG_OPEN_IDENTIFIER |
| **Message** | BGP Open message received with bad identifier %s |
| **Description** | BGP Open message received with bad identifier |
| **Log Module** | bgp |

| | |
|---|---|
| **Log Group** | peer |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_MSG_OPEN_HOLDTIMER |
| **Message** | BGP peer %s, source %s, hostname %s, instance %s received open message with illegal hold timer %s |
| **Description** | BGP Open message received with bad hold timer value |
| **Log Module** | bgp |
| **Log Group** | peer |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_INVALID_RECEIVED_AS |
| **Message** | BGP peer %s, source %s, hostname %s, instance %s received wrong AS %s, expected AS %s |
| **Description** | BGP Received wrong AS number from remote peer |
| **Log Module** | bgp |
| **Log Group** | peer |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_MSG_MANDATORY_ATTRIBUTE |
| **Message** | BGP peer %s, source %s, hostname %s, instance %s received message %s with mandatory parameter %s missing |
| **Description** | BGP message mandatory parameter missing |
| **Log Module** | bgp |
| **Log Group** | peer |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_PEER_RESET_MSG_HDR_LEN_ERR |
| **Message** | BGP peer %s, source %s, hostname %s, instance %s is reset, reason: invalid message header length %s |
| **Description** | BGP peer reset log for invalid header length |
| **Log Module** | bgp |
| **Log Group** | peer |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_PEER_RESET_INVALID_TYPE |
| **Message** | BGP peer %s, source %s, hostname %s, instance %s is reset, reason: invalid message type %s, length %s |
| **Description** | BGP peer reset log for invalid message type |
| **Log Module** | bgp |
| **Log Group** | peer |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_PEER_RESET_INVALID_PARAMETER_TYPE |
| **Message** | BGP peer %s, source %s, hostname %s, instance %s is reset, reason: invalid optional parameter, message %u, parameter type %u |
| **Description** | BGP peer reset log for invalid optional parameter type |
| **Log Module** | bgp |
| **Log Group** | peer |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_PEER_RESET_INVALID_CAP_LEN |

| | |
|---|---|
| **Message** | BGP peer %s, source %s hostname %s, instance %s is reset, reason: invalid capability type %s, capability length %s |
| **Description** | BGP peer reset log for invalid capability length |
| **Log Module** | bgp |
| **Log Group** | peer |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_DUPLICATE_TCP_ACCEPT |
| **Message** | BGP peer %s, source %s, hostname %s, instance %s duplicate accept socket is success. |
| **Description** | BGP Duplicate accept socket success |
| **Log Module** | bgp |
| **Log Group** | peer |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_IFL_CHANGE |
| **Message** | Interface %s, instance %s received event %s |
| **Description** | Interface status change event |
| **Log Module** | bgp |
| **Log Group** | interface |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_IFA_CHANGE |
| **Message** | Interface %s, instance %s received %s of address %s |
| **Description** | Interface address event |
| **Log Module** | bgp |
| **Log Group** | interface |

| | |
|---|---|
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_INSTANCE_INACTIVE |
| **Message** | Instance %s is inactive, reason %s |
| **Description** | BGP Instance is inactive |
| **Log Module** | bgp |
| **Log Group** | instance |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_MSG_VALIDATION_FAIL |
| **Message** | BGP message received from peer %s, source %s, hostname %s, instance %s validation failed. Message type %s |
| **Description** | BGP Message validation failure |
| **Log Module** | bgp |
| **Log Group** | message |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_MSG_DECODE_FAIL |
| **Message** | BGP message received from peer %s, source %s, hostname %s, instance %s decode failed. Message type %s |
| **Description** | BGP Message decode failure |
| **Log Module** | bgp |
| **Log Group** | message |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_UPD_SET_ERR_INFO |
| **Message** | Update processing error. code %s , subcode %s |
| **Description** | BGP Update processing error code and subcode |
| **Log Module** | bgp |
| **Log Group** | message |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_UPD_INVALID_IPV4_PREFIX |
| **Message** | BGP update message processing, peer %s, source %s, hostname %s, instance %s: Received invalid prefix %s/%s |
| **Description** | BGP Update message received with invalid prefix |
| **Log Module** | bgp |
| **Log Group** | message |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_UPD_ATTR_REPEAT |
| **Message** | BGP update message processing, peer %s, source %s, hostname %s, instance %s: Attribute %s repeated |
| **Description** | BGP Update message with attribute repeated |
| **Log Module** | bgp |
| **Log Group** | message |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_UPD_ATTR_WELLKNOWN_TRANS |
| **Message** | BGP update message processing, peer %s, source %s, hostname %s, instance %s: Attribute %s does not have well known transitive flag set |
| **Description** | BGP Update processing error: Attribute does not have well known transitive flag set |

| | |
|---|---|
| **Log Module** | bgp |
| **Log Group** | message |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_UPD_ATTR_WK_TRANS_PARTIAL |
| **Message** | BGP update message processing, peer %s, source %s, hostname %s, instance %s: Well known transitive attribute %s has partial bit set |
| **Description** | BGP Update processing error: Well known transitive attribute has partial bit set |
| **Log Module** | bgp |
| **Log Group** | message |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_UPD_ATTR_INVALID_ASPATH_LEN |
| **Message** | BGP update message processing, peer %s, source %s, hostname %s, instance %s: Invalid AS path length %s |
| **Description** | BGP Update processing: Invalid AS path length |
| **Log Module** | bgp |
| **Log Group** | message |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_UPD_ENFORCE_FIRST_AS |
| **Message** | BGP update message processing, peer %s, source %s, hostname %s, instance %s: First AS %s is not peer AS %s, ignoring |
| **Description** | BGP Update processing: First AS is not peer AS, so ignore the update |
| **Log Module** | bgp |
| **Log Group** | message |
| **Severity** | Info |

| | |
|---|---|
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_UPD_INVALID_SEG_TYPE |
| **Message** | BGP update message processing, peer %s, source %s, hostname %s, instance %s: Invalid path segment type %s |
| **Description** | BGP Update processing: Invalid segment type |
| **Log Module** | bgp |
| **Log Group** | message |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_UPD_INVALID_NH |
| **Message** | BGP update message processing, peer %s, source %s, hostname %s, instance %s: Invalid unicast ip address in nexthop %s |
| **Description** | BGP Update processing: Invalid unicast ip address in nexthop |
| **Log Module** | bgp |
| **Log Group** | message |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_UPD_INVALID_IP6_PREFIX |
| **Message** | BGP update message processing, peer %s, source %s, hostname %s, instance %s: Invalid prefix %s/%s |
| **Description** | BGP Update processing Invalid ipv6 prefix |
| **Log Module** | bgp |
| **Log Group** | message |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_UPD_RCVD_NON_NEGOTIATED_AF |
| **Message** | BGP update message processing, peer %s, source %s, hostname %s, instance %s: Received NLRI of non-negotiated address family %s, safi %s |
| **Description** | BGP Update processing: Received NLRI of non-negotiated address family |
| **Log Module** | bgp |
| **Log Group** | message |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_UPD_INVALID_LEN |
| **Message** | BGP update message processing, peer %s, source %s, hostname %s, instance %s: Received invalid length %s |
| **Description** | BGP Update processing: Received invalid length |
| **Log Module** | bgp |
| **Log Group** | message |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_UPD_MALFORMED_PATH_ATTR |
| **Message** | BGP update message processing, peer %s, source %s, hostname %s, instance %s: Received malformed path attribute %s, length %s |
| **Description** | BGP Update processing: Received malformed path attribute |
| **Log Module** | bgp |
| **Log Group** | message |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_UPD_ATTR_EXT_LEN |
| **Message** | BGP update message processing, peer %s, source %s, hostname %s, instance %s: Invalid length bit set for attribute %s |
| **Description** | BGP Update processing: Invalid length bit set for attribute |

| | |
|---|---|
| **Log Module** | bgp |
| **Log Group** | message |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_UPD_ATTR_INVALID_ORIGIN |
| **Message** | BGP update message processing, peer %s, source %s, hostname %s, instance %s: Invalid origin attribute value %s |
| **Description** | BGP Update processing: Invalid origin attribute value |
| **Log Module** | bgp |
| **Log Group** | message |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_UPD_ATTR_OPTIONAL_TRANS |
| **Message** | BGP update message processing, peer %s, source %s, hostname %s, instance %s: Attribute %s does not have optional transitive flag set |
| **Description** | BGP Update processing: Attribute does not have optional transitive flag set |
| **Log Module** | bgp |
| **Log Group** | message |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_UPD_OWN_ROUTER_ID |
| **Message** | BGP update message processing, peer %s, source %s, hostname %s, instance %s: Received own router id %s |
| **Description** | BGP Update processing: Received own router ID |
| **Log Module** | bgp |
| **Log Group** | message |
| **Severity** | Info |

| | |
|---|---|
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_UPD_ATTR_OPTIONAL_NON_TRANS |
| **Message** | BGP update message processing, peer %s, source %s, hostname %s, instance %s: Attribute %s does not have optional non-transitive flag set |
| **Description** | BGP Update processing: Attribute does not have optional non-transitive flag set |
| **Log Module** | bgp |
| **Log Group** | message |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_UPD_INVALID_ATTR_EBGP |
| **Message** | BGP update message processing, peer %s, source %s, hostname %s, instance %s: Received invalid attribute %s from EBGP peer |
| **Description** | BGP Update processing: Received invalid attribute from EBGP peer |
| **Log Module** | bgp |
| **Log Group** | message |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_UPD_INVALID_SR_IDX_LEN |
| **Message** | BGP update message processing, peer %s, source %s, hostname %s, instance %s: Invalid SR index length %s |
| **Description** | BGP Update processing: Invalid SR index length |
| **Log Module** | bgp |
| **Log Group** | message |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_UPD_INVALID_SRGB_LEN |
| **Message** | BGP update message processing, peer %s, source %s, hostname %s, instance %s: Invalid SRGB length %s |
| **Description** | BGP Update processing: Invalid SRGB length |
| **Log Module** | bgp |
| **Log Group** | message |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_UPD_PATH_ATTR_FAIL |
| **Message** | BGP update message processing, peer %s, source %s, hostname %s, instance %s: Processing path attribute failed |
| **Description** | BGP Update processing: Processing path attribute failed |
| **Log Module** | bgp |
| **Log Group** | message |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_UPD_MREACH_FAIL |
| **Message** | BGP update message processing, peer %s, source %s, hostname %s, instance %s: Processing MPREACH NLRI attribute failed |
| **Description** | BGP Update processing: Processing MPREACH NLRI attribute failed |
| **Log Module** | bgp |
| **Log Group** | message |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_UPD_INVALID_EXP_LEN |
| **Message** | BGP update message processing, peer %s, source %s, hostname %s, instance %s: Received invalid length %s . Expected %s |
| **Description** | BGP Update processing: Received invalid length |

| | |
|---|---|
| **Log Module** | bgp |
| **Log Group** | message |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_UPD_NLRI_FAIL |
| **Message** | BGP update message processing, peer %s, source %s, hostname %s, instance %s: Processing NLRI attribute failed |
| **Description** | BGP Update processing: Processing NLRI attribute failed |
| **Log Module** | bgp |
| **Log Group** | message |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_FSM_PEER_STATE_CHANGE |
| **Message** | Peer [%s] Src[%s] CurState[%s] Event[%s] NextState[%s] |
| **Description** | BGP Update processing: Processing NLRI attribute failed |
| **Log Module** | bgp |
| **Log Group** | message |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_UPD_FLOWSPEC_INVALID_LEN |
| **Message** | BGP flowspec update message processing, peer %s, source %s, hostname %s, instance %s: Received invalid flowspec nlri(%s) |
| **Description** | BGP Update processing: Received invalid length |
| **Log Module** | bgp |
| **Log Group** | message |
| **Severity** | Info |

| | |
|---|---|
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_NO_ROUTER_ID |
| **Message** | Router ID is not present in instance %s |
| **Description** | BGP Router ID is present |
| **Log Module** | bgp |
| **Log Group** | peer |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_NO_LOCAL_AS |
| **Message** | Local AS is not configured in instance %s |
| **Description** | BGP Local AS is not configured |
| **Log Module** | bgp |
| **Log Group** | peer |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_FLOWSPEC_RECEIVED |
| **Message** | Flowspec rule %s received from Peer(peer:%s, src:%s), flowspec hash (%s) |
| **Description** | BGP Flowspec update/withdraw received |
| **Log Module** | bgp |
| **Log Group** | flowspec |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_FLOWSPEC_SEND |

| | |
|---|---|
| **Message** | Flowspec rule %s send to Peer(peer:%s, src:%s), (flowspec hash: %s) |
| **Description** | BGP Flowspec send to peer |
| **Log Module** | bgp |
| **Log Group** | flowspec |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_FLOWSPEC_RULE_INVALID |
| **Message** | Flowspec rule (flowspec hash: %s) not updated |
| **Description** | BGP Flowspec not updated |
| **Log Module** | bgp |
| **Log Group** | flowspec |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_FLOWSPEC_ENTRY_DELETED |
| **Message** | Flowspec rule (flowspec hash: %s) deleted |
| **Description** | BGP Flowspec rule deleted |
| **Log Module** | bgp |
| **Log Group** | flowspec |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_FLOWSPEC_PRIORITY_ADDED |
| **Message** | Flowspec rule (flowspec hash: %s) added at index %s |
| **Description** | BGP Flowspec rule priority added |
| **Log Module** | bgp |
| **Log Group** | flowspec |
| **Severity** | debug |

| | |
|---|---|
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_FLOWSPEC_ACL_ADDED |
| **Message** | Flowspec rule (flowspec hash: %s) added with priority %s and acl count %s |
| **Description** | BGP Flowspec rule acl added |
| **Log Module** | bgp |
| **Log Group** | flowspec |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_FLOWSPEC_ACL_DELETED |
| **Message** | Flowspec rule (flowspec hash: %s) deleted with priority %s and acl count %s |
| **Description** | BGP Flowspec rule acl deleted |
| **Log Module** | bgp |
| **Log Group** | flowspec |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_FLOWSPEC_ACL_REINSTALLED |
| **Message** | Flowspec rule (flowspec hash: %s) reinstalled with priority %s and acl count %s |
| **Description** | BGP Flowspec rule acl reinstalled |
| **Log Module** | bgp |
| **Log Group** | flowspec |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_FLOWSPEC_PRIORITY_DELETED |
| **Message** | Flowspec rule (flowspec hash: %s) at index %s deleted |
| **Description** | BGP Flowspec rule priority deleted |
| **Log Module** | bgp |
| **Log Group** | flowspec |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_FLOWSPEC_PRIORITY_UPDATED |
| **Message** | Flowspec rule (flowspec hash: %s) priority updated from %s to %s acl-full:%s status:%s |
| **Description** | BGP Flowspec rule priority updated |
| **Log Module** | bgp |
| **Log Group** | flowspec |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_FLOWSPEC_PRIORITY_FULL |
| **Message** | Flowspec rule (flowspec hash: %s) not sorted as the max rule count %s reached |
| **Description** | BGP Flowspec rule not sorted |
| **Log Module** | bgp |
| **Log Group** | flowspec |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_FLOWSPEC_PRIORITY_NOT_FOUND |
| **Message** | Flowspec rule (flowspec hash: %s) not found in priority sorting |
| **Description** | BGP Flowspec rule not found |
| **Log Module** | bgp |

| | |
|---|---|
| **Log Group** | flowspec |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_FLOWSPEC_PRIORITY_FOUND |
| **Message** | Flowspec rule (flowspec hash: %s) found in priority sorting at index %s |
| **Description** | BGP Flowspec rule found |
| **Log Module** | bgp |
| **Log Group** | flowspec |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | BGP_FLOWSPEC_STATUS_INVALID |
| **Message** | Status invalid flowspec(%s) |
| **Description** | Invalid status |
| **Log Module** | bgp |
| **Log Group** | flowspec |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

## nat Log Messages

#### Log Level Summary

Warning(1), Notice(3)

| | |
|---|---|
| **Log ID** | NAT_LOG_POOL_RESOURCE_STATUS |
| **Message** | Pool exhaustion, status %s, pool %s, instance %s |
| **Description** | Resource Usage Status |
| **Log Module** | nat |
| **Log Group** | nat-resource |
| **Severity** | warning |

| | |
|---|---|
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | NAT_LOG_USER_ADDR_PORT_ALLOC |
| **Message** | IPv4 Address Allocated, User %s, Original Address %s, Translated Address %s, L4-Port-Range %s - %s |
| **Description** | User IPv4 Address and Port-Range Allocated |
| **Log Module** | nat |
| **Log Group** | nat-resource |
| **Severity** | notice |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | NAT_LOG_USER_ADDR_PORT_FREE |
| **Message** | IPv4 Address Freed, User %s, Original Address %s, Translated Address %s, L4-Port-Range %s - %s |
| **Description** | User IPv4 Address and Port-Range Freed |
| **Log Module** | nat |
| **Log Group** | nat-resource |
| **Severity** | notice |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | NAT_LOG_RULE_USAGE_STATUS |
| **Message** | Max rules, status %s, subscriber %s |
| **Description** | Address Translation Rules Usage of a Subsciber |
| **Log Module** | nat |
| **Log Group** | nat-resource |
| **Severity** | notice |
| **Recommended Actions** | Contact Rtbrick customer support |

## rtbrick-cli Log Messages

### Log Level Summary

Info(1)

| | |
|---|---|
| **Log ID** | CLI_BACKEND_COMMAND_EXECUTION_LOG |
| **Message** | %s |
| **Description** | Command executed via cli/restconf |
| **Log Module** | rtbrick-cli |
| **Log Group** | cli-execution |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

## hostconfd Log Messages

### Log Level Summary

Error(5), Info(6)

| | |
|---|---|
| **Log ID** | HOSTCONFD_ALERTING_CONFIG_ERROR |
| **Message** | HOSTCONFD alert config error {event:%s} |
| **Description** | Error reported by hostconfd on alerting configuration error |
| **Log Module** | hostconfd |
| **Log Group** | functional |
| **Severity** | ERROR |
| **Recommended Actions** | Check the hostconfd operational status |

| | |
|---|---|
| **Log ID** | IPMI_CONFIG_TABLE_ERROR |
| **Message** | Failed to create IPMI config table %s |
| **Description** | Failed to create IPMI config table |
| **Log Module** | hostconfd |
| **Log Group** | functional |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | WATCHDOG_CONFIG_TABLE_ERROR |
| **Message** | Failed to create Watchdog config table %s |
| **Description** | Failed to create Watchdog config table |
| **Log Module** | hostconfd |
| **Log Group** | functional |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SNMP_CONFIG_TABLE_ERROR |
| **Message** | Failed to create snmp config table %s |
| **Description** | Failed to create snmp config table |
| **Log Module** | hostconfd |
| **Log Group** | functional |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | FAIL2BAN_CONFIG_TABLE_ERROR |
| **Message** | Failed to create fail2ban config table %s |
| **Description** | Failed to create fail2ban config table |
| **Log Module** | hostconfd |
| **Log Group** | functional |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | HOSTCONFD_BDS_GENERIC_FAIL |
| **Message** | Failed due to bds events- {table:%s} |
| **Description** | Failed due to bds events |
| **Log Module** | hostconfd |

| | |
|---|---|
| **Log Group** | bds |
| **Severity** | INFO |
| **Recommended Actions** | Check table/object in the bds commands |

| | |
|---|---|
| **Log ID** | HOSTCONFD_TABLE_CONFIG_READ_FAIL |
| **Message** | Failed to read static hostconfd table configuration- {file:%s} |
| **Description** | Failed to read static hostconfd table configuration |
| **Log Module** | hostconfd |
| **Log Group** | config |
| **Severity** | INFO |
| **Recommended Actions** | Check the json format/table-name in the file |

| | |
|---|---|
| **Log ID** | HOSTCONFD_DYN_PLUGIN_REGN_FAIL |
| **Message** | Failed to register dynamic plugin- {table-type:%s} |
| **Description** | Failed to register dynamic plugin |
| **Log Module** | hostconfd |
| **Log Group** | config |
| **Severity** | INFO |
| **Recommended Actions** | Check the table definition |

| | |
|---|---|
| **Log ID** | HOSTCONFD_HTTP_HEADER_FAIL |
| **Message** | Failed to create HTTP header to send to hostconfd {event:%s} |
| **Description** | Failed to create HTTP header to send to hostconfd |
| **Log Module** | hostconfd |
| **Log Group** | functional |
| **Severity** | INFO |
| **Recommended Actions** | Check the table definition |

| | |
|---|---|
| **Log ID** | HOSTCONFD_HTTP_CONTENT_FAIL |
| **Message** | Failed to create HTTP content to send to hostconfd- {table-type:%s} |
| **Description** | Failed to create HTTP content to send to hostconfd |
| **Log Module** | hostconfd |
| **Log Group** | functional |
| **Severity** | INFO |
| **Recommended Actions** | Check the table deifinition |

| | |
|---|---|
| **Log ID** | HOSTCONFD_CONNECTION_DOWN |
| **Message** | HOSTCONFD connection is DOWN {event:%s} |
| **Description** | HOSTCONFD connection is DOWN |
| **Log Module** | hostconfd |
| **Log Group** | functional |
| **Severity** | INFO |
| **Recommended Actions** | Check the hostconfd operational status |

### Interfaces Log Messages

#### Log Level Summary

Info(20)

| | |
|---|---|
| **Log ID** | IFM_OPER_IFP_ADMIN_UP |
| **Message** | Admin status of the physical interface %s is UP |
| **Description** | Interface configuration change triggered |
| **Log Module** | ifm |
| **Log Group** | config |
| **Severity** | info |
| **Recommended Actions** | Check the interface configurations for this physical interface |

| | |
|---|---|
| **Log ID** | IFM_OPER_IFP_ADMIN_DOWN |
| **Message** | Admin status of the physical interface %s is DOWN |

| | |
|---|---|
| **Description** | Interface configuration change triggered |
| **Log Module** | ifm |
| **Log Group** | config |
| **Severity** | info |
| **Recommended Actions** | Check the interface configurations for this physical interface |

| | |
|---|---|
| **Log ID** | IFM_OPER_IFP_LINK_UP |
| **Message** | Link status of the physical interface %s is UP |
| **Description** | Interface link events triggered |
| **Log Module** | ifm |
| **Log Group** | interface-events |
| **Severity** | info |
| **Recommended Actions** | Check the physical connection for this particular port |

| | |
|---|---|
| **Log ID** | IFM_OPER_IFP_LINK_DOWN |
| **Message** | Link status of the physical interface %s is DOWN |
| **Description** | Interface link events triggered |
| **Log Module** | ifm |
| **Log Group** | interface-events |
| **Severity** | info |
| **Recommended Actions** | Check the physical connection for this particular port |

| | |
|---|---|
| **Log ID** | IFM_OPER_IFP_OPERATIONAL_UP |
| **Message** | Operational status of the physical interface %s is UP |
| **Description** | Interface config or link events triggered |
| **Log Module** | ifm |
| **Log Group** | interface-events |
| **Severity** | info |
| **Recommended Actions** | Check the physical connection for this particular port or the interface configuration |

| | |
|---|---|
| **Log ID** | IFM_OPER_IFP_OPERATIONAL_DOWN |
| **Message** | Operational status of the physical interface %s is DOWN |
| **Description** | Interface config or link events triggered |
| **Log Module** | ifm |
| **Log Group** | interface-events |
| **Severity** | info |
| **Recommended Actions** | Check the physical connection for this particular port or the interface configuration |

| | |
|---|---|
| **Log ID** | IFM_OPER_IFL_ADMIN_UP |
| **Message** | Admin status of the logical interface %s is UP |
| **Description** | Interface configuration change triggered |
| **Log Module** | ifm |
| **Log Group** | config |
| **Severity** | info |
| **Recommended Actions** | Check the interface configurations for this logical interface |

| | |
|---|---|
| **Log ID** | IFM_OPER_IFL_ADMIN_DOWN |
| **Message** | Admin status of the logical interface %s is DOWN |
| **Description** | Interface configuration change triggered |
| **Log Module** | ifm |
| **Log Group** | config |
| **Severity** | info |
| **Recommended Actions** | Check the interface configurations for this logical interface |

| | |
|---|---|
| **Log ID** | IFM_OPER_IFL_LINK_UP |
| **Message** | Link status of the logical interface %s is UP |
| **Description** | Interface link events triggered |
| **Log Module** | ifm |

| | |
|---|---|
| **Log Group** | interface-events |
| **Severity** | info |
| **Recommended Actions** | Check the physical connection for this particular port |

| | |
|---|---|
| **Log ID** | IFM_OPER_IFL_LINK_DOWN |
| **Message** | The link status of the logical interface %s is DOWN |
| **Description** | Interface link events triggered |
| **Log Module** | ifm |
| **Log Group** | interface-events |
| **Severity** | info |
| **Recommended Actions** | Check the physical connection for this particular port |

| | |
|---|---|
| **Log ID** | IFM_OPER_IFL_OPERATIONAL_UP |
| **Message** | Operational status of the logical interface %s is UP |
| **Description** | Interface config or link events triggered |
| **Log Module** | ifm |
| **Log Group** | interface-events |
| **Severity** | info |
| **Recommended Actions** | Check the physical connection for this particular port or the logical interface configuration |

| | |
|---|---|
| **Log ID** | IFM_OPER_IFL_OPERATIONAL_DOWN |
| **Message** | The operational status of the logical interface %s is DOWN |
| **Description** | Interface config or link events triggered |
| **Log Module** | ifm |
| **Log Group** | interface-events |
| **Severity** | info |
| **Recommended Actions** | Check the physical connection for this particular port or the logical interface configuration |

| | |
|---|---|
| **Log ID** | IFM_OPER_IFL_IPV4_STATUS_UP |
| **Message** | IPv4 status of the logical interface %s is UP |
| **Description** | Logical Interface configuration change triggered |
| **Log Module** | ifm |
| **Log Group** | config |
| **Severity** | info |
| **Recommended Actions** | Check the ipv4 status for this logical interface |

| | |
|---|---|
| **Log ID** | IFM_OPER_IFL_IPV4_STATUS_DOWN |
| **Message** | IPv4 status of the logical interface %s is DOWN |
| **Description** | Logical Interface configuration change triggered |
| **Log Module** | ifm |
| **Log Group** | config |
| **Severity** | info |
| **Recommended Actions** | Check the ipv4 status for this logical interface |

| | |
|---|---|
| **Log ID** | IFM_OPER_IFL_IPV6_STATUS_UP |
| **Message** | IPv6 status of the logical interface %s is UP |
| **Description** | Logical Interface configuration change triggered |
| **Log Module** | ifm |
| **Log Group** | config |
| **Severity** | info |
| **Recommended Actions** | Check the ipv6 status for this logical interface |

| | |
|---|---|
| **Log ID** | IFM_OPER_IFL_IPV6_STATUS_DOWN |
| **Message** | IPv6 status of the logical interface %s is DOWN |
| **Description** | Logical Interface configuration change triggered |
| **Log Module** | ifm |
| **Log Group** | config |
| **Severity** | info |

| | |
|---|---|
| **Recommended Actions** | Check the ipv6 status for this logical interface |

| | |
|---|---|
| **Log ID** | IFM_OPER_IFL_MPLS_STATUS_UP |
| **Message** | The MPLS status of the logical interface %s is UP |
| **Description** | Logical Interface configuration change triggered |
| **Log Module** | ifm |
| **Log Group** | config |
| **Severity** | info |
| **Recommended Actions** | Check the mpls status for this logical interface |

| | |
|---|---|
| **Log ID** | IFM_OPER_IFL_MPLS_STATUS_DOWN |
| **Message** | MPLS status of the logical interface %s is DOWN |
| **Description** | Logical Interface configuration change triggered |
| **Log Module** | ifm |
| **Log Group** | config |
| **Severity** | info |
| **Recommended Actions** | Check the mpls status for this logical interface |

| | |
|---|---|
| **Log ID** | IFM_OPER_IFP_HOLDDOWN_START |
| **Message** | The holddown machinery for the physical interface %s is activated |
| **Description** | Interface config or link events triggered |
| **Log Module** | ifm |
| **Log Group** | interface-events |
| **Severity** | info |
| **Recommended Actions** | Check the physical connection for this particular port or its configuration |

| | |
|---|---|
| **Log ID** | IFM_OPER_IFP_HOLDDOWN_STOP |

| | |
|---|---|
| **Message** | The holddown machinery for the physical interface %s is deactivated after %s carrier transitions |
| **Description** | Interface config or link events triggered |
| **Log Module** | ifm |
| **Log Group** | interface-events |
| **Severity** | info |
| **Recommended Actions** | Check the physical connection for this particular port or its configuration |

## IPoE Log Messages

### Log Level Summary

Info(17), Debug(24)

| | |
|---|---|
| **Log ID** | SUBSCRIBER_CREATED |
| **Message** | IPoE subscriber %s created for %s.%s:%s %s |
| **Description** | IPoE subscriber created |
| **Log Module** | ipoe |
| **Log Group** | subscriber |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSCRIBER_DELETED |
| **Message** | IPoE subscriber %s deleted with terminate code %s |
| **Description** | IPoE subscriber deleted |
| **Log Module** | ipoe |
| **Log Group** | subscriber |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCP_BINDING_CREATED |
| **Message** | IPoE subscriber %s DHCP binding created |

| | |
|---|---|
| **Description** | Subscriber DHCP binding created |
| **Log Module** | ipoe |
| **Log Group** | dhcp |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCP_BINDING_DELETED |
| **Message** | IPoE subscriber %s DHCP binding deleted |
| **Description** | Subscriber DHCP binding deleted |
| **Log Module** | ipoe |
| **Log Group** | dhcp |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCP_BINDING_EXPIRED |
| **Message** | IPoE subscriber %s DHCP binding lease timer expired |
| **Description** | Subscriber DHCP binding lease expired |
| **Log Module** | ipoe |
| **Log Group** | dhcp |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCPV6_BINDING_CREATED |
| **Message** | IPoE subscriber %s DHCPv6 binding created |
| **Description** | Subscriber DHCPv6 binding created |
| **Log Module** | ipoe |
| **Log Group** | dhcp |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCPV6_BINDING_DELETED |
| **Message** | IPoE subscriber %s DHCPv6 lease expired |
| **Description** | Subscriber DHCPv6 binding deleted |
| **Log Module** | ipoe |
| **Log Group** | dhcp |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCPV6_BINDING_EXPIRED |
| **Message** | IPoE subscriber %s DHCPv6 binding lease timer expired |
| **Description** | Subscriber DHCPv6 binding lease expired |
| **Log Module** | ipoe |
| **Log Group** | dhcp |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCP_OPTION_IGNORED |
| **Message** | IPoE subscriber %s %s option %s ignored (%s) |
| **Description** | Subscriber DHCP option ignored |
| **Log Module** | ipoe |
| **Log Group** | dhcp |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCP_BINDING_RELEASED |
| **Message** | IPoE subscriber %s DHCP binding released |
| **Description** | Subscriber DHCP binding released |
| **Log Module** | ipoe |

| | |
|---|---|
| **Log Group** | dhcp |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCP_BINDING_FAILED |
| **Message** | IPoE subscriber %s DHCP binding failed |
| **Description** | Subscriber DHCP binding lease expired |
| **Log Module** | ipoe |
| **Log Group** | dhcp |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCPV6_BINDING_RELEASED |
| **Message** | IPoE subscriber %s DHCPv6 binding released |
| **Description** | Subscriber DHCPv6 binding released |
| **Log Module** | ipoe |
| **Log Group** | dhcp |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCPV6_BINDING_FAILED |
| **Message** | IPoE subscriber %s DHCPv6 binding failed |
| **Description** | Subscriber DHCPv6 binding failed |
| **Log Module** | ipoe |
| **Log Group** | dhcp |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSCRIBER_STATE |
| **Message** | IPoE subscriber %s state changed from %s to %s |
| **Description** | IPoE subscriber state |
| **Log Module** | ipoe |
| **Log Group** | subscriber |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCPV6_IA_STATUS |
| **Message** | IPoE subscriber %s DHCPv6 %s from %s status code %s (%s) |
| **Description** | Subscriber DHCPv6 IA status code received |
| **Log Module** | ipoe |
| **Log Group** | dhcp |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCP_RELAY_TRAP |
| **Message** | Trap rule %s %s for DHCP relay |
| **Description** | DHCP relay trap installation |
| **Log Module** | ipoe |
| **Log Group** | dhcp-relay |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCP_RELAY_SHM_DECODE |
| **Message** | DHCP relay packet shm validation error |
| **Description** | Failed to decode received DHCP relay packet |
| **Log Module** | ipoe |
| **Log Group** | dhcp-relay |
| **Severity** | info |

| | |
|---|---|
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSCRIBER_DELETE_FAILED |
| **Message** | IPoE subscriber %s deletion with terminate code %s failed |
| **Description** | IPoE subscriber deletion failed |
| **Log Module** | ipoe |
| **Log Group** | subscriber |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSCRIBER_RD_FAILURE |
| **Message** | IPoE subscriber %s from redundancy session %s failed due to %s |
| **Description** | IPoE redundancy subscriber failed |
| **Log Module** | ipoe |
| **Log Group** | redundancy |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | IPOED_BDS_TABLE_CREATE_ERROR |
| **Message** | Failed to create BDS table %s |
| **Description** | Failed to create BDS table |
| **Log Module** | ipoe |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCP_RELAY_DECODE_ERROR |

| | |
|---|---|
| **Message** | DHCP relay packet decode error on interface %s, error code: %s subcode: %s |
| **Description** | Failed to decode received DHCP relay packet |
| **Log Module** | ipoe |
| **Log Group** | dhcp-relay |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCP_RELAY_IP_UDP_DECODE |
| **Message** | DHCP relay packet IP header decode error on interface %s |
| **Description** | Failed to decode received DHCP relay packet |
| **Log Module** | ipoe |
| **Log Group** | dhcp-relay |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCP_RELAY_GATEWAY_IP |
| **Message** | DHCP relay gateway IP address not configured for interface %s |
| **Description** | DHCP relay config missing |
| **Log Module** | ipoe |
| **Log Group** | dhcp-relay |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCP_RELAY_BINDING_ADD |
| **Message** | DHCP relay binding added for client %s on interface %s with gateway IP %s, client-ip %s and lease-time %s |
| **Description** | DHCP relay binding added |
| **Log Module** | ipoe |
| **Log Group** | dhcp-relay |

| | |
|---|---|
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCP_RELAY_BINDING_DELETE |
| **Message** | DHCP relay binding deleted for client %s on interface %s |
| **Description** | DHCP relay binding deleted |
| **Log Module** | ipoe |
| **Log Group** | dhcp-relay |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCP_RELAY_BINDING_MISSING |
| **Message** | DHCP relay binding missing for client %s, ignoring the packet from server |
| **Description** | DHCP relay binding missing |
| **Log Module** | ipoe |
| **Log Group** | dhcp-relay |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCP_RELAY_CONFIG |
| **Message** | DHCP relay configuration missing for interface %s |
| **Description** | DHCP relay config missing |
| **Log Module** | ipoe |
| **Log Group** | dhcp-relay |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCP_RELAY_SERVER_LIST_CONFIG |

| | |
|---|---|
| **Message** | DHCP relay server configuration missing on interface %s, dropping the DHCP packet %s |
| **Description** | DHCP server config missing |
| **Log Module** | ipoe |
| **Log Group** | dhcp-relay |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCP_RELAY_SERVER_CONFIG |
| **Message** | DHCP relay server IP configuration missing on interface %s for DHCP server %s |
| **Description** | DHCP server config missing |
| **Log Module** | ipoe |
| **Log Group** | dhcp-relay |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCP_RELAY_SERVER_GET |
| **Message** | Unable to get DHCP relay server %s configuration on interface %s |
| **Description** | DHCP server config missing |
| **Log Module** | ipoe |
| **Log Group** | dhcp-relay |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCP_RELAY_PACKET_CLIENT_RCV |
| **Message** | DHCP %s received from client on interface %s with MAC %s |
| **Description** | DHCP relay packet received from client |
| **Log Module** | ipoe |
| **Log Group** | dhcp-relay |

| | |
|---|---|
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCP_RELAY_PACKET_SERVER_RCV |
| **Message** | DHCP %s received from server %s for client on interface %s with MAC %s |
| **Description** | DHCP relay packet received from server |
| **Log Module** | ipoe |
| **Log Group** | dhcp-relay |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCP_RELAY_PACKET_CLIENT_SENT |
| **Message** | DHCP %s from server %s sent to client on interface %s with MAC %s |
| **Description** | DHCP relay packet sent to client |
| **Log Module** | ipoe |
| **Log Group** | dhcp-relay |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCP_RELAY_PACKET_SERVER_SENT |
| **Message** | DHCP %s sent to server %s from client on interface %s with MAC %s |
| **Description** | DHCP relay packet sent to server |
| **Log Module** | ipoe |
| **Log Group** | dhcp-relay |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCP_RELAY_PACKET_CLIENT_RCV_INVALID |

| | |
|---|---|
| **Message** | Invalid DHCP %s received from client on interface %s with MAC %s |
| **Description** | DHCP relay packet received from client |
| **Log Module** | ipoe |
| **Log Group** | dhcp-relay |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCP_RELAY_PACKET_SERVER_RCV_INVALID |
| **Message** | Invalid DHCP %s received from server %s |
| **Description** | DHCP relay packet received from server |
| **Log Module** | ipoe |
| **Log Group** | dhcp-relay |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCP_RELAY_PROXY_SERVER |
| **Message** | Selecting the server %s for the client on interface %s in proxy mode |
| **Description** | DHCP relay proxy server selected |
| **Log Module** | ipoe |
| **Log Group** | dhcp-relay |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCP_RELAY_PROXY_IGNORE_SERVER_PACKET |
| **Message** | Ignoring packet %s from server %s as the selected server is %s |
| **Description** | DHCP relay proxy packet ignored from server |
| **Log Module** | ipoe |
| **Log Group** | dhcp-relay |
| **Severity** | debug |

| | |
|---|---|
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCP_RELAY_ENCODE_ERROR |
| **Message** | Error in encoding DHCP %s packet with client interface %s with MAC %s |
| **Description** | DHCP relay encode error |
| **Log Module** | ipoe |
| **Log Group** | dhcp-relay |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCP_RELAY_PROXY_NO_SERVER |
| **Message** | Server identifier missing in DHCP relay binding, dropping the DHCP %s packet from client interface %s with MAC %s |
| **Description** | DHCP relay proxy no server in binding |
| **Log Module** | ipoe |
| **Log Group** | dhcp-relay |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | DHCP_RELAY_INVALID_DST_IP |
| **Message** | Invalid destination IP, dropping the DHCP %s packet from client interface %s with MAC %s |
| **Description** | DHCP relay proxy no server in binding |
| **Log Module** | ipoe |
| **Log Group** | dhcp-relay |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

## IS-IS Log Messages

### Log Level Summary

Alert(1), Info(14), Debug(4)

| | |
|---|---|
| **Log ID** | ISIS_AUTHENTICATION_FAIL |
| **Message** | Authentication failure for PDU %s, interface %s, instance %s, level %s |
| **Description** | ISIS Authentication failure log |
| **Log Module** | isis |
| **Log Group** | message |
| **Severity** | alert |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | ISIS_INSTANCE_OVERLOAD |
| **Message** | ISIS instance %s, level %s, %s, overload state |
| **Description** | ISIS instance overload enter/exit log |
| **Log Module** | isis |
| **Log Group** | instance |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | ISIS_NEIGHBOR_DISCOVERY |
| **Message** | ISIS neighbor %s, discovered, interface %s, instance %s |
| **Description** | ISIS neighbor discovery log |
| **Log Module** | isis |
| **Log Group** | neighbor |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | ISIS_NEIGHBOR_STATE_CHANGE |

| | |
|---|---|
| **Message** | ISIS neighbor %s, interface %s, instance %s, changed state from %s, to %s, reason %s |
| **Description** | ISIS neighbor FSM state change log |
| **Log Module** | isis |
| **Log Group** | neighbor |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | ISIS_IFL_CHANGE |
| **Message** | Interface %s, instance %s, received event %s, operational state %s, admin state %s, ipv4 status %s, ipv6 status %s |
| **Description** | ISIS enabled interface status change event |
| **Log Module** | isis |
| **Log Group** | interface |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | ISIS_IFA_CHANGE |
| **Message** | Interface %s, instance %s, received %s, for address %s |
| **Description** | ISIS enabled interface address event |
| **Log Module** | isis |
| **Log Group** | interface |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | ISIS_PDU_DECODE_FAIL |
| **Message** | ISIS PDU %s, decode failure received in interface %s, instance %s, level %s, reason %s |
| **Description** | ISIS PDU decode failure log |
| **Log Module** | isis |

| | |
|---|---|
| **Log Group** | message |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | ISIS_LSP_FLOOD_FILTER_BLOCK |
| **Message** | LSP flooding is blocked for neighbor %s, in interface %s, instance %s, level %s, for lsp-id %s |
| **Description** | ISIS flood filter block log |
| **Log Module** | isis |
| **Log Group** | message |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | ISIS_SPF_COMPUTATION |
| **Message** | Dijkstra's Shortest Path First algorithm triggered in instance %s, level %s. Reason %s |
| **Description** | Dijkstra's Shortest Path First algorithm triggered |
| **Log Module** | isis |
| **Log Group** | spf |
| **Severity** | Info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | ISIS_SPF_COMPUTATION_LSP |
| **Message** | Dijkstra's Shortest Path First algorithm triggered in instance %s, level %s. Reason LSP %s, received |
| **Description** | LSP received triggers Dijkstra's Shortest Path First algorithm |
| **Log Module** | isis |
| **Log Group** | spf |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | ISIS_ORIGINATE_SELF_LSP |
| **Message** | Self lsp is originate in instance %s, level %s, lsp id %s, sequence number %s |
| **Description** | Self lsp origination log |
| **Log Module** | isis |
| **Log Group** | lsp |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | ISIS_IPV4_PREFIX_ADD |
| **Message** | Add %s, metric %s, source %s, preference %s, instance %s, topology %s |
| **Description** | ipv4 prefix addition log |
| **Log Module** | isis |
| **Log Group** | route |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | ISIS_IPV6_PREFIX_ADD |
| **Message** | Add %s, metric %s, source %s, preference %s, instance %s, topology %s |
| **Description** | ipv6 prefix addition log |
| **Log Module** | isis |
| **Log Group** | route |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | ISIS_MPLS_LABEL_ADD |
| **Message** | Add mpls label %s, source %s, preference %s, instance %s, topology %s |
| **Description** | MPLS prefix addition log |
| **Log Module** | isis |

| | |
|---|---|
| **Log Group** | route |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | ISIS_MPLS_LABEL_DELETE |
| **Message** | Delete mpls-label %s, instance %s, topology %s |
| **Description** | MPLS prefix deletion log |
| **Log Module** | isis |
| **Log Group** | route |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | ISIS_NEIGHBOR_DELETED |
| **Message** | ISIS neighbor %s, deleted, interface %s, instance %s |
| **Description** | ISIS neighbor deleted log |
| **Log Module** | isis |
| **Log Group** | neighbor |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | ISIS_ORIGINATE_LSP_PURGE |
| **Message** | Purge lsp is originated in instance %s, level %s, lsp id %s, sequence number %s |
| **Description** | Purge lsp origination log |
| **Log Module** | isis |
| **Log Group** | lsp |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | ISIS_IPV4_PREFIX_DELETE |
| **Message** | Delete %s, instance %s, topology %s |
| **Description** | ipv4 prefix deletion log |
| **Log Module** | isis |
| **Log Group** | route |
| **Severity** | Debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | ISIS_IPV6_PREFIX_DELETE |
| **Message** | Delete %s, instance %s, topology %s |
| **Description** | ipv6 prefix deletion log |
| **Log Module** | isis |
| **Log Group** | route |
| **Severity** | Debug |
| **Recommended Actions** | Contact Rtbrick customer support |

## LDP Log Messages

### Log Level Summary

Info(5), Debug(18)

| | |
|---|---|
| **Log ID** | LDP_NBR_CREATED |
| **Message** | LDP Neighbour created, peer %s, interface %s, instance %s |
| **Description** | LDP has formed a new neighbor |
| **Log Module** | ldp |
| **Log Group** | adjacency |
| **Severity** | INFO |
| **Recommended Actions** | None Required. |

| | |
|---|---|
| **Log ID** | LDP_NBR_DELETED |
| **Message** | LDP Neighbour deleted, peer %s, interface %s, instance %s |

| | |
|---|---|
| **Description** | LDP neighbor is removed. |
| **Log Module** | ldp |
| **Log Group** | adjacency |
| **Severity** | INFO |
| **Recommended Actions** | Configure the neighbor router again for establishing neighborship and sending hello packets. |

| | |
|---|---|
| **Log ID** | LDP_NOTIFICATION_IN |
| **Message** | LDP Notification received, peer %s, instance %s, reason %s |
| **Description** | LDP has received notification messages. There are basically two types of notification, one is advisory and the other is fatal. In case a fatal notification is received, the session will be reset. |
| **Log Module** | ldp |
| **Log Group** | notification |
| **Severity** | INFO |
| **Recommended Actions** | If the peer is reset, run the `show ldp session detail` command and check the “Reason” attribute. |

| | |
|---|---|
| **Log ID** | LDP_NOTIFICATION_OUT |
| **Message** | LDP Notification sent, peer %s, instance %s, reason %s |
| **Description** | LDP has sent out notification messages to the peer. |
| **Log Module** | ldp |
| **Log Group** | notification |
| **Severity** | INFO |
| **Recommended Actions** | If peer is reset, run the `show ldp session detail` command and check the “Reason” attribute. |

| | |
|---|---|
| **Log ID** | LDP_SESSION_STATE |
| **Message** | LDP session state changed to %s, peer %s, instance %s |
| **Description** | LDP session state has changed |
| **Log Module** | ldp |
| **Log Group** | peer |
| **Severity** | INFO |

| | |
|---|---|
| **Recommended Actions** | None Required. |

| | |
|---|---|
| **Log ID** | LDP_SESSION_CREATED |
| **Message** | LDP Session created, peer %s, instance %s |
| **Description** | LDP has formed a new session. |
| **Log Module** | ldp |
| **Log Group** | peer |
| **Severity** | debug |
| **Recommended Actions** | None Required. |

| | |
|---|---|
| **Log ID** | LDP_SESSION_DELETED |
| **Message** | LDP Session deleted, peer %s, instance %s |
| **Description** | LDP session is deleted. |
| **Log Module** | ldp |
| **Log Group** | peer |
| **Severity** | debug |
| **Recommended Actions** | Run the `show ldp session detail` command and check the “Reason” attribute for information. |

| | |
|---|---|
| **Log ID** | LDP_TCP_STATE_CHNG |
| **Message** | LDP TCP state changed from "%s" to "%s" due to event "%s", peer %s, instance %s |
| **Description** | LDP TCP state has changed. |
| **Log Module** | ldp |
| **Log Group** | peer |
| **Severity** | debug |
| **Recommended Actions** | None Required. |

| | |
|---|---|
| **Log ID** | LDP_SESSION_STATE_CHNG |

| | |
|---|---|
| **Message** | LDP Session state changed from "%s" to "%s" due to event "%s", peer %s, instance %s |
| **Description** | LDP session state has changed. |
| **Log Module** | ldp |
| **Log Group** | peer |
| **Severity** | debug |
| **Recommended Actions** | None Required. |

| | |
|---|---|
| **Log ID** | LDP_MSG_ADDR_OUT |
| **Message** | LDP Address message sent for prefix %s, peer %s, instance %s |
| **Description** | LDP address message is sent. |
| **Log Module** | ldp |
| **Log Group** | peer |
| **Severity** | debug |
| **Recommended Actions** | None Required. |

| | |
|---|---|
| **Log ID** | LDP_MSG_ADDR_IN |
| **Message** | LDP Address message received for prefix %s, peer %s, instance %s |
| **Description** | LDP address message is received. |
| **Log Module** | ldp |
| **Log Group** | peer |
| **Severity** | debug |
| **Recommended Actions** | None Required. |

| | |
|---|---|
| **Log ID** | LDP_MSG_ADDR_WITHDRAW_OUT |
| **Message** | LDP Address withdraw message sent for prefix %s, peer %s, instance %s |
| **Description** | LDP address withdrawal message is sent. |
| **Log Module** | ldp |
| **Log Group** | peer |
| **Severity** | debug |

| | |
|---|---|
| **Recommended Actions** | None Required. |

| | |
|---|---|
| **Log ID** | LDP_MSG_ADDR_WITHDRAW_IN |
| **Message** | LDP Address withdraw message received for prefix %s, peer %s, instance %s |
| **Description** | LDP address withdrawal message is received. |
| **Log Module** | ldp |
| **Log Group** | peer |
| **Severity** | debug |
| **Recommended Actions** | None Required. |

| | |
|---|---|
| **Log ID** | LDP_MSG_LABEL_MAP_OUT |
| **Message** | LDP label mapping message sent for prefix %s with label-%s, peer %s, instance %s |
| **Description** | LDP mapping message is sent. |
| **Log Module** | ldp |
| **Log Group** | peer |
| **Severity** | debug |
| **Recommended Actions** | None Required. |

| | |
|---|---|
| **Log ID** | LDP_MSG_LABEL_MAP_IN |
| **Message** | LDP label mapping message received for prefix %s with label-%s, peer %s, instance %s |
| **Description** | LDP mapping is received. |
| **Log Module** | ldp |
| **Log Group** | peer |
| **Severity** | debug |
| **Recommended Actions** | None Required. |

| | |
|---|---|
| **Log ID** | LDP_MSG_LABEL_WITHDRAW_OUT |
| **Message** | LDP label withdraw message sent for prefix %s with label-%s, peer %s, instance %s |
| **Description** | LDP mapping withdrawal message is sent. |
| **Log Module** | ldp |
| **Log Group** | peer |
| **Severity** | debug |
| **Recommended Actions** | None Required. |

| | |
|---|---|
| **Log ID** | LDP_MSG_LABEL_WITHDRAW_IN |
| **Message** | LDP label withdraw message received for prefix %s with label-%s, peer %s, instance %s |
| **Description** | LDP mapping withdrawal message is received. |
| **Log Module** | ldp |
| **Log Group** | peer |
| **Severity** | debug |
| **Recommended Actions** | None Required. |

| | |
|---|---|
| **Log ID** | LDP_MSG_LABEL_REQ_REPLY_OUT |
| **Message** | LDP label request message reply for prefix %s sent with label-%s, peer %s, instance %s |
| **Description** | LDP request reply message is sent. |
| **Log Module** | ldp |
| **Log Group** | peer |
| **Severity** | debug |
| **Recommended Actions** | None Required. |

| | |
|---|---|
| **Log ID** | LDP_MSG_LABEL_REQ_IN |
| **Message** | LDP label request message received, prefix %s, peer %s, instance %s |
| **Description** | LDP request message is received. |

| | |
|---|---|
| **Log Module** | ldp |
| **Log Group** | peer |
| **Severity** | debug |
| **Recommended Actions** | None Required. |

| | |
|---|---|
| **Log ID** | LDP_MSG_LABEL_REL_OUT |
| **Message** | LDP label release message sent, prefix %s, label-%s, peer %s, instance %s |
| **Description** | LDP release message is sent. |
| **Log Module** | ldp |
| **Log Group** | peer |
| **Severity** | debug |
| **Recommended Actions** | None Required. |

| | |
|---|---|
| **Log ID** | LDP_MSG_LABEL_REL_IN |
| **Message** | LDP label release message received, prefix %s, label-%s, peer %s, instance %s |
| **Description** | LDP release message is received. |
| **Log Module** | ldp |
| **Log Group** | peer |
| **Severity** | debug |
| **Recommended Actions** | None Required. |

| | |
|---|---|
| **Log ID** | LDP_ADD_TO_FIB |
| **Message** | LDP route added to FIB, prefix %s, label-%s, instance %s |
| **Description** | LDP has been added to FIB. |
| **Log Module** | ldp |
| **Log Group** | peer |
| **Severity** | debug |
| **Recommended Actions** | None Required. |

| | |
|---|---|
| **Log ID** | LDP_DEL_FROM_FIB |
| **Message** | LDP route deleted from FIB, prefix %s, label-%s, instance %s |
| **Description** | LDP is deleted from FIB. |
| **Log Module** | ldp |
| **Log Group** | peer |
| **Severity** | debug |
| **Recommended Actions** | None Required. |

## License Log Messages

### Log Level Summary

Critical(4), Error(16), Warning(1), Notice(1), Info(2)

| | |
|---|---|
| **Log ID** | LICENSE_GRACE_PERIOD_ERROR |
| **Message** | The current license has %s days remaining in its grace period |
| **Description** | License active |
| **Log Module** | license |
| **Log Group** | operational |
| **Severity** | ALERT |
| **Recommended Actions** | Please install a new license |

| | |
|---|---|
| **Log ID** | LICENSE_TRAIL_PERIOD_EXPIRY |
| **Message** | The software trial period has ended, please install a license |
| **Description** | The software trial period has ended, please install a license |
| **Log Module** | license |
| **Log Group** | operational |
| **Severity** | CRITICAL |
| **Recommended Actions** | Please install a license |

| | |
|---|---|
| **Log ID** | LICENSE_EXPIRY_CRITICAL |
| **Message** | The current license has expired, please install a new license |
| **Description** | The current license has expired |
| **Log Module** | license |
| **Log Group** | operational |
| **Severity** | CRITICAL |
| **Recommended Actions** | Please install a new license |

| | |
|---|---|
| **Log ID** | LICENSE_INSTALL_FAIL |
| **Message** | License installation failed. Please try again. If its persists contact support |
| **Description** | The license has not been installed |
| **Log Module** | license |
| **Log Group** | operational |
| **Severity** | CRITICAL |
| **Recommended Actions** | Please contact RtBrick support |

| | |
|---|---|
| **Log ID** | LICENSE_EXPIRED_INSTALL |
| **Message** | License installed has already expired, please install a valid license |
| **Description** | The current license has expired |
| **Log Module** | license |
| **Log Group** | operational |
| **Severity** | CRITICAL |
| **Recommended Actions** | Please install a new license |

| | |
|---|---|
| **Log ID** | LICENSE_INTERNAL_NULL_STRING |
| **Message** | NULL license string passed to license verification API |
| **Description** | Invalid parameters passed for license verification API |
| **Log Module** | license |
| **Log Group** | internal |
| **Severity** | ERROR |

| | |
|---|---|
| **Recommended Actions** | Internal error, please contact Rtbrick Customer Support |

| | |
|---|---|
| **Log ID** | LICENSE_FILE_READ_FAILURE |
| **Message** | Unable to read license string from license file %s |
| **Description** | License file read failed |
| **Log Module** | license |
| **Log Group** | internal |
| **Severity** | ERROR |
| **Recommended Actions** | Internal error, please contact Rtbrick Customer Support |

| | |
|---|---|
| **Log ID** | LICENSE_FILE_OPEN_FAILURE |
| **Message** | Unable to open from license file %s |
| **Description** | Unable to open license file |
| **Log Module** | license |
| **Log Group** | internal |
| **Severity** | ERROR |
| **Recommended Actions** | Internal error, please contact Rtbrick Customer Support |

| | |
|---|---|
| **Log ID** | LICENSE_STRING_ALLOCATION_FAILURE |
| **Message** | Unable to allocate %s bytes for license string |
| **Description** | Memory allocation failure |
| **Log Module** | license |
| **Log Group** | internal |
| **Severity** | ERROR |
| **Recommended Actions** | Internal error, please contact Rtbrick Customer Support |

| | |
|---|---|
| **Log ID** | LICENSE_INVALID_STRING_IN_FILE |
| **Message** | Invalid license string present in license file %s |

| | |
|---|---|
| **Description** | Invalid license installed |
| **Log Module** | license |
| **Log Group** | internal |
| **Severity** | ERROR |
| **Recommended Actions** | Internal error, please contact Rtbrick Customer Support |

| | |
|---|---|
| **Log ID** | LICENSE_INVALID_CONTENT |
| **Message** | Invalid license content in file %s |
| **Description** | Invalid license content |
| **Log Module** | license |
| **Log Group** | internal |
| **Severity** | ERROR |
| **Recommended Actions** | Internal error, please contact Rtbrick Customer Support |

| | |
|---|---|
| **Log ID** | LICENSE_DIRECTORY_OPEN_FAILURE |
| **Message** | Unable to open license directory |
| **Description** | License directory open failure |
| **Log Module** | license |
| **Log Group** | internal |
| **Severity** | ERROR |
| **Recommended Actions** | Internal error, please contact Rtbrick Customer Support |

| | |
|---|---|
| **Log ID** | LICENSE_START_DATE_MISSING |
| **Message** | License doesn’t have a start date |
| **Description** | License doesn’t have a start date |
| **Log Module** | license |
| **Log Group** | internal |
| **Severity** | ERROR |
| **Recommended Actions** | Start date missing in licence please ask for a new license |

| | |
|---|---|
| **Log ID** | LICENSE_END_DATE_MISSING |
| **Message** | License doesn’t have a end date |
| **Description** | License doesn’t have a end date |
| **Log Module** | license |
| **Log Group** | internal |
| **Severity** | ERROR |
| **Recommended Actions** | End date missing in license please ask for a new license |

| | |
|---|---|
| **Log ID** | LICENSE_DIRECTORY_CLOSE_FAILURE |
| **Message** | Unable to close license directory |
| **Description** | Unable to close license directory |
| **Log Module** | license |
| **Log Group** | internal |
| **Severity** | ERROR |
| **Recommended Actions** | Internal error, please contact Rtbrick Customer Support |

| | |
|---|---|
| **Log ID** | LICENSE_INIT_FAILED |
| **Message** | License initialization failed |
| **Description** | License initialization failed |
| **Log Module** | license |
| **Log Group** | internal |
| **Severity** | ERROR |
| **Recommended Actions** | Internal error, please contact Rtbrick Customer Support |

| | |
|---|---|
| **Log ID** | LICENSE_TABLE_NOT_PRESENT |
| **Message** | License table (global.rtbrick.license) is not present |
| **Description** | License table (global.rtbrick.license) is not present |
| **Log Module** | license |

| | |
|---|---|
| **Log Group** | internal |
| **Severity** | ERROR |
| **Recommended Actions** | Internal error, please contact Rtbrick Customer Support |

| | |
|---|---|
| **Log ID** | LICENSE_SECONDARY_VALIDATION_COMPROMISED |
| **Message** | License table (global.rtbrick.license) is not present |
| **Description** | License table (global.rtbrick.license) is not present |
| **Log Module** | license |
| **Log Group** | internal |
| **Severity** | ERROR |
| **Recommended Actions** | Internal error, please contact Rtbrick Customer Support |

| | |
|---|---|
| **Log ID** | LICENSE_QUOTA_TABLE_CREATE_FAILED |
| **Message** | License quota table (secure.license.quota) is not present |
| **Description** | License quota table (secure.license.quota) is not present |
| **Log Module** | license |
| **Log Group** | internal |
| **Severity** | ERROR |
| **Recommended Actions** | Internal error, please contact Rtbrick Customer Support |

| | |
|---|---|
| **Log ID** | LICENSE_CONFIG_TABLE_CREATE_FAILED |
| **Message** | License config table (local.rtbrick.license.config) is not present |
| **Description** | License config table (local.rtbrick.license.config) is not present |
| **Log Module** | license |
| **Log Group** | internal |
| **Severity** | ERROR |
| **Recommended Actions** | Internal error, please contact Rtbrick Customer Support |

| | |
|---|---|
| **Log ID** | LICENSE_EXPIRY_ERROR |
| **Message** | The current license will expire today, please install a new license |
| **Description** | The current license about to expire |
| **Log Module** | license |
| **Log Group** | operational |
| **Severity** | ERROR |
| **Recommended Actions** | Please install a new license |

| | |
|---|---|
| **Log ID** | LICENSE_EXPIRY_WARNING |
| **Message** | The current license will expire in %s day(s), please install a new license |
| **Description** | The current license about to expire |
| **Log Module** | license |
| **Log Group** | operational |
| **Severity** | WARNING |
| **Recommended Actions** | Please install a new license |

| | |
|---|---|
| **Log ID** | LICENSE_INSTALL_SUCCESSFUL |
| **Message** | Valid license has been successfully installed |
| **Description** | License active |
| **Log Module** | license |
| **Log Group** | operational |
| **Severity** | NOTICE |
| **Recommended Actions** | Device is up for operation |

| | |
|---|---|
| **Log ID** | LICENSE_INTERNAL_INVALID_STRING |
| **Message** | Invalid license string passed to license verification API |
| **Description** | Invalid parameters passed for license verification API |
| **Log Module** | license |
| **Log Group** | internal |
| **Severity** | INFO |

| | |
|---|---|
| **Recommended Actions** | Internal error, please contact Rtbrick Customer Support |

| | |
|---|---|
| **Log ID** | LICENSE_HOSTCONFD_CONNECTION_DOWN |
| **Message** | License installation delayed due to hostconfd connection down |
| **Description** | License installation delayed due to hostconfd connection down |
| **Log Module** | license |
| **Log Group** | internal |
| **Severity** | INFO |
| **Recommended Actions** | Internal error, please contact Rtbrick Customer Support |

## Policy Log Messages

### Log Level Summary

Error(3), Warning(1), Info(3)

| | |
|---|---|
| **Log ID** | POLICY_LOG_CODE_GEN_ERROR |
| **Message** | Policy [%s] Term [%s] Match Type [%s] List [%s] Command [%s] Ordinal [%s], SubOrdinal [%s] : Error [%s] |
| **Description** | Generation Policy errors |
| **Log Module** | policy |
| **Log Group** | Generation |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | POLICY_LOG_CODE_GEN_STATUS_FAILED |
| **Message** | Policy [%s]: Code Generation Error Reason [%s] |
| **Description** | Generation Policy Error Reason |
| **Log Module** | policy |
| **Log Group** | Generation |
| **Severity** | ERROR |

| | |
|---|---|
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | POLICY_LOG_BDS_ERROR |
| **Message** | Policy [%s]: Error [%s] |
| **Description** | Policy & BDS API interaction errors |
| **Log Module** | policy |
| **Log Group** | BDS |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | POLICY_LOG_CONFIG_PROC_WARNING |
| **Message** | Policy [%s]: Warning [%s] |
| **Description** | Policy Configuration processing warnings |
| **Log Module** | policy |
| **Log Group** | Configuration |
| **Severity** | WARNING |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | POLICY_LOG_CODE_GEN_EVENTS |
| **Message** | Policy [%s] Term [%s] Match Type [%s] List [%s] Command [%s] Ordinal [%s], SubOrdinal [%s] : Event [%s] |
| **Description** | Generation Policy Events |
| **Log Module** | policy |
| **Log Group** | Generation |
| **Severity** | INFO |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | POLICY_LOG_CODE_GEN_STATUS_SUCCESS |

| | |
|---|---|
| **Message** | Policy [%s]: Code Generation Status: [%s] |
| **Description** | Generation Policy Status |
| **Log Module** | policy |
| **Log Group** | Generation |
| **Severity** | INFO |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | POLICY_LOG_CODE_GEN_NOTIFICATION |
| **Message** | Policy [%s]: Notification Status [%s] |
| **Description** | Generation Policy Notification |
| **Log Module** | policy |
| **Log Group** | Generation |
| **Severity** | INFO |
| **Recommended Actions** | Contact Rtbrick customer support |

### PPPoE Log Messages

#### Log Level Summary

Notice(1), Info(35), Debug(21)

| | |
|---|---|
| **Log ID** | PPPOE_SESSION_LIMIT_REACHED |
| **Message** | PPPoE session limit reached on interface %s, MAC %s, S-Tag %s, C-Tag %s |
| **Description** | PPPoE session limit reached |
| **Log Module** | pppoe |
| **Log Group** | pppoe |
| **Severity** | notice |
| **Recommended Actions** | New session can be created only after removing existing sessions |

| | |
|---|---|
| **Log ID** | PPPOE_CONFIG_NOT_FOUND |
| **Message** | No matching configuration found for PPPoE discovery on interface %s, MAC %s, S-Tag %s, C-Tag %s |

| | |
|---|---|
| **Description** | Received PPPoE discovery packet without matching configuration |
| **Log Module** | pppoe |
| **Log Group** | config |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_SUBSCRIBER_ID_GET_FAILED |
| **Message** | PPPoE session subscriber-id get failed on interface %s, MAC %s, PPPoE session-id %s, S-Tag %s, C-Tag %s |
| **Description** | Failed to assign subscriber identifier to PPPoE session |
| **Log Module** | pppoe |
| **Log Group** | subscriber |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_FSM_STATE_CHANGE |
| **Message** | %s session for subscriber-id %s changed state from %s to %s for input %s, action %s |
| **Description** | PPP FSM state changed |
| **Log Module** | pppoe |
| **Log Group** | ppp |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_PPP_SEND_ERROR |
| **Message** | Failed to encode and send PPP packet on interface %s, subscriber-id %s |
| **Description** | Failed to encode and send PPP packet |
| **Log Module** | pppoe |
| **Log Group** | ppp |
| **Severity** | info |

| | |
|---|---|
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_LCP_WRONG_ECHO_MAGIC |
| **Message** | LCP ECHO received on interface %s, subscriber-id %s with wrong magic number %s, expected %s |
| **Description** | PPP LCP echo request/reply received with wrong magic number |
| **Log Module** | pppoe |
| **Log Group** | ppp |
| **Severity** | info |
| **Recommended Actions** | Check the magic number |

| | |
|---|---|
| **Log ID** | PPPOE_LCP_WRONG_ECHO_REPLY_ID |
| **Message** | LCP ECHO reply received on interface %s, subscriber-id %s with wrong identifier %s, expected %s |
| **Description** | PPP LCP echo reply received with wrong identifier |
| **Log Module** | pppoe |
| **Log Group** | ppp |
| **Severity** | info |
| **Recommended Actions** | Check LCP packet identifier |

| | |
|---|---|
| **Log ID** | PPPOE_LCP_RCV_PROTO_REJECT |
| **Message** | PPP protocol reject received on interface %s, subscriber-id %s protocol %s |
| **Description** | PPP protocol reject received |
| **Log Module** | pppoe |
| **Log Group** | ppp |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_PROTO_DECODE_ERROR |
| **Message** | %s packet decode error on interface %s, MAC %s, session-id %s, subscriber-id %s with remaining packet length %s |
| **Description** | Failed to decode received PPPoE PPP packet |
| **Log Module** | pppoe |
| **Log Group** | ppp |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_LCP_RCV_WRONG_ID |
| **Message** | LCP %s received on interface %s, subscriber-id %s with identifier %s, expected %s |
| **Description** | PPP LCP response with wrong identifer received |
| **Log Module** | pppoe |
| **Log Group** | ppp |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_LCP_RCV_MRU |
| **Message** | LCP %s received on interface %s, subscriber-id %s with received MRU %s, local MRU %s |
| **Description** | PPP LCP response with wrong or suggested MRU received |
| **Log Module** | pppoe |
| **Log Group** | ppp |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_LCP_RCV_MAGIC |
| **Message** | LCP %s received on interface %s, subscriber-id %s with received magic %s, local magic %s |
| **Description** | PPP LCP response with wrong or suggested magic number received |

| | |
|---|---|
| **Log Module** | pppoe |
| **Log Group** | ppp |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_LCP_RCV_AUTH |
| **Message** | LCP %s received on interface %s, subscriber-id %s with wrong authentication protocol %s, expected %s |
| **Description** | PPP LCP response with wrong or suggested auth protocol received |
| **Log Module** | pppoe |
| **Log Group** | ppp |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_PPP_RCV_ERROR |
| **Message** | %s receive error %s on interface %s, MAC %s, session-id %s, subscriber-id %s |
| **Description** | PPP receive errors |
| **Log Module** | pppoe |
| **Log Group** | ppp |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_PROTO_MAX_RETRANSMISSION |
| **Message** | Maximum %s retransmission count on interface %s, subscriber-id %s, retransmit count %s |
| **Description** | PPPoE PPP retransmission exceeded |
| **Log Module** | pppoe |
| **Log Group** | ppp |
| **Severity** | info |

| | |
|---|---|
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_AUTH_REJECT |
| **Message** | Authentication rejected on interface %s, subscriber-id %s with user %s |
| **Description** | PPPoE PPP authentication rejected |
| **Log Module** | pppoe |
| **Log Group** | ppp |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_NO_SOURCE_IP |
| **Message** | %s source IP address not found on interface %s, subscriber-id %s |
| **Description** | Missing source IP address to setup PPPoE PPP session |
| **Log Module** | pppoe |
| **Log Group** | ppp |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_PROTO_RECEIVE_ERROR |
| **Message** | Protocol receive error %s on interface %s, MAC %s, session-id %s, subscriber-id %s |
| **Description** | PPPoE PPP protocol receive error |
| **Log Module** | pppoe |
| **Log Group** | ppp |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_PROTO_RCV_ID_WARNING |

| | |
|---|---|
| **Message** | %s %s received on interface %s, subscriber-id %s with wrong identifier %s, expected %s |
| **Description** | PPP NCP reply received with wrong identifier |
| **Log Module** | pppoe |
| **Log Group** | ppp |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_IPCP_RECEIVE_WRONG_IP |
| **Message** | IPCP %s received on interface %s, subscriber-id %s with address %s, expected %s |
| **Description** | PPP IPCP reply received with wrong address |
| **Log Module** | pppoe |
| **Log Group** | ppp |
| **Severity** | info |
| **Recommended Actions** | Check the IP address configured in the client |

| | |
|---|---|
| **Log ID** | PPPOE_IPV6_CHECKSUM_ERROR |
| **Message** | PPPoE IPv6 packet decode checksum error on interface %s, subscriber-id %s |
| **Description** | Failed to decode received PPPoE PPP packet with wrong IPv6 checksum (UDP or ICMPv6) |
| **Log Module** | pppoe |
| **Log Group** | ppp |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_DECODE_ERROR |
| **Message** | Error decoding received PPPoE packet on interface %s, MAC %s |
| **Description** | Failed to decode PPPoE ethertype or VLAN |
| **Log Module** | pppoe |

| | |
|---|---|
| **Log Group** | pppoe |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_DECODE_ERROR_WITH_VLAN |
| **Message** | Error decoding received PPPoE packet on interface %s, MAC %s, S-Tag %s, C-Tag %s |
| **Description** | Failed to decode PPPoE discovery packet |
| **Log Module** | pppoe |
| **Log Group** | pppoe |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_UNKNOWN_DISCOVERY_PACKET |
| **Message** | Received unknown PPPoE discovery packet on interface %s, MAC %s, S-Tag %s, C-Tag %s, session-id %s |
| **Description** | Received unknown PPPoE discovery packet |
| **Log Module** | pppoe |
| **Log Group** | pppoe |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_SESSION_CREATE |
| **Message** | PPPoE session created on interface %s, S-Tag %s, C-Tag %s, MAC %s, PPPoE session-id %s, subscriber-id %s, state %s access-profile %s |
| **Description** | Create of PPPoE/PPP session after PADS is sent |
| **Log Module** | pppoe |
| **Log Group** | pppoe |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_PACKET_ENCODE_ERROR |
| **Message** | Could not encode PPPoE %s on interface %s, MAC %s, PPPoE session-id %s, S-Tag %s, C-Tag %s |
| **Description** | PPPoE discovery encode error |
| **Log Module** | pppoe |
| **Log Group** | pppoe |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_BAD_PACKET_INTERFACE_GET_FAILED |
| **Message** | Interface get failed for PPPoE %s packet on interface %s, MAC %s |
| **Description** | PPPoE discovery packet received on invalid interface |
| **Log Module** | pppoe |
| **Log Group** | pppoe |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_BAD_PACKET_INTERFACE_DOWN |
| **Message** | Received PPPoE %s packet on operationaly down interface %s, MAC %s |
| **Description** | PPPoE discovery packet received on down interface |
| **Log Module** | pppoe |
| **Log Group** | pppoe |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_BAD_PACKET_WRONG_MAC |
| **Message** | Received bad PPPoE %s packet with wrong MAC on interface %s, MAC %s |
| **Description** | Invalid PPPoE discovery packet received with wrong MAC |
| **Log Module** | pppoe |

| | |
|---|---|
| **Log Group** | pppoe |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_BAD_PACKET_NO_AC_COOKIE |
| **Message** | Received bad PPPoE %s packet on interface %s, MAC %s with no ac cookie |
| **Description** | Invalid PPPoE discovery packet received with no ac_cookie |
| **Log Module** | pppoe |
| **Log Group** | pppoe |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_BAD_PACKET_WRONG_AC_COOKIE |
| **Message** | Received bad PPPoE %s packet on interface %s, MAC %s with wrong ac cookie |
| **Description** | Invalid PPPoE discovery packet received with wrong ac_cookie |
| **Log Module** | pppoe |
| **Log Group** | pppoe |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_BAD_PACKET_UNKNOWN_ETHER_TYPE |
| **Message** | Received bad PPPoE packet on interface %s, MAC %s with unknown ether type %s |
| **Description** | Received PPPoE discovery packet with invalid ethertype |
| **Log Module** | pppoe |
| **Log Group** | pppoe |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_SESSION_DELETE |
| **Message** | PPPoE session delete on interface %s, MAC %s, PPPoE session-id %s, S-Tag %s, C-Tag %s |
| **Description** | PPPoE session delete |
| **Log Module** | pppoe |
| **Log Group** | pppoe |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_DHCPV6_DECODE_ERROR |
| **Message** | PPPoE DHCPv6 packet decode error on interface %s, subscriber-id %s error code: %s subcode: %s |
| **Description** | Failed to decode received PPPoE DHCPv6 packet |
| **Log Module** | pppoe |
| **Log Group** | dhcpv6 |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_DHCPV6_VALIDATION_ERROR |
| **Message** | PPPoE DHCPv6 packet validation error on interface %s, subscriber-id %s |
| **Description** | Failed to validate received PPPoE DHCPv6 packet |
| **Log Module** | pppoe |
| **Log Group** | dhcpv6 |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_VENDOR_REQUEST_FAILED |
| **Message** | LCP vendor request failed for subscriber-id %s, %s |
| **Description** | PPPoE PPP LCP vendor request failed |

| | |
|---|---|
| **Log Module** | pppoe |
| **Log Group** | ppp |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_RECEIVE_ERROR |
| **Message** | PPPOE packet receive with %s |
| **Description** | Generic PPPoE receive error |
| **Log Module** | pppoe |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_CONFIG_FOUND |
| **Message** | PPPoE %s received on interface %s, MAC %s, S-Tag %s, C-Tag %s, found access-profile %s |
| **Description** | Received PPPoE discovery packet with matching configuration |
| **Log Module** | pppoe |
| **Log Group** | config |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_SUBSCRIBER_IN |
| **Message** | Result from subscriberd for subscriber-id %s with action %s |
| **Description** | Action received from subscriberd |
| **Log Module** | pppoe |
| **Log Group** | subscriber |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_LCP_SESSION_EVENT |
| **Message** | LCP session %s on interface %s, MAC %s, PPPoE session-id %s, subscriber-id %s |
| **Description** | PPP LCP protocol events |
| **Log Module** | pppoe |
| **Log Group** | ppp |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_IPCP_SESSION_EVENT |
| **Message** | IPCP session %s on interface %s, MAC %s, PPPoE session-id %s, subscriber-id %s |
| **Description** | PPP IPCP protocol events |
| **Log Module** | pppoe |
| **Log Group** | ppp |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_IP6CP_SESSION_EVENT |
| **Message** | IP6CP session %s on interface %s, MAC %s, PPPoE session-id %s, subscriber-id %s |
| **Description** | PPP IP6CP protocol events |
| **Log Module** | pppoe |
| **Log Group** | ppp |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_LCP_PRINT_MY_OPTIONS |

| | |
|---|---|
| **Message** | PPP LCP options for subscriber-id %s, MRU %s, magic %s, authentication %s, identifier %s, retransmission-count %s, retransmisson-interval %s |
| **Description** | Local PPP LCP options |
| **Log Module** | pppoe |
| **Log Group** | ppp |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_LCP_START_ECHO |
| **Message** | LCP start echo on interface %s, subscriber-id %s |
| **Description** | Start sending PPP LCP echo request interval |
| **Log Module** | pppoe |
| **Log Group** | ppp |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_LCP_RCV_CONFX |
| **Message** | PPP LCP %s received on interface %s, subscriber-id %s, MRU %s, magic %s, authentication %s |
| **Description** | PPP LCP request/response received |
| **Log Module** | pppoe |
| **Log Group** | ppp |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_LCP_NEGOTIATION |
| **Message** | LCP %s on interface %s for subscriber-id %s |
| **Description** | PPP LCP negotiation |
| **Log Module** | pppoe |
| **Log Group** | ppp |

| | |
|---|---|
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_LCP_SETUP |
| **Message** | LCP setup %s on interface %s for subscriber-id %s |
| **Description** | PPPoE PPP sessions setup |
| **Log Module** | pppoe |
| **Log Group** | ppp |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_AUTH_RCV_REQ |
| **Message** | %s receive request packet on interface %s, subscriber-id %s, user %s |
| **Description** | PPPoE PPP authentication request received |
| **Log Module** | pppoe |
| **Log Group** | ppp |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_LCP_UPPPER_LAYER_TEARDOWN |
| **Message** | LCP teardown by upper layer on interface %s for subscriber-id %s |
| **Description** | PPPoE PPP session teardown/termination |
| **Log Module** | pppoe |
| **Log Group** | ppp |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_NCP_SESSION_DETAIL |

| | |
|---|---|
| **Message** | %s session on interface %s, subscriber-id %s, retransmission-interval %s, local ipv4 address %s |
| **Description** | PPPoE PPP NCP local session info |
| **Log Module** | pppoe |
| **Log Group** | ppp |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_IPCP_CONFREQ_PEER |
| **Message** | IPCP CONFREQ peer info for subscriber-id %s on interface %s, peer ipv4-address %s, primary dns %s, secondary dns %s |
| **Description** | PPPoE PPP IPCP session info |
| **Log Module** | pppoe |
| **Log Group** | ppp |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_IPCP_CONFREQ |
| **Message** | IPCP CONFREQ received for subscriber-id %s on interface %s, peer ipv4-address %s, primary dns %s, secondary dns %s |
| **Description** | PPPoE PPP IPCP received |
| **Log Module** | pppoe |
| **Log Group** | ppp |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_DISCOVERY_PACKET_RECEIVE |
| **Message** | Received PPPoE %s packet on interface %s, MAC %s, session-id %s |
| **Description** | Received PPPoE discovery packet |
| **Log Module** | pppoe |

| | |
|---|---|
| **Log Group** | pppoe |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_DISCOVERY_PACKET_SEND |
| **Message** | Sending PPPoE %s packet on interface %s, MAC %s, session-id %s |
| **Description** | Send PPPoE discovery packet |
| **Log Module** | pppoe |
| **Log Group** | pppoe |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_DHCPV6_LEASE |
| **Message** | PPPoE DHCPv6 IA_PD %s for subscriber-id %s, MAC %s, S-Tag %s, C-Tag %s |
| **Description** | PPPoE DHCPv6 IA_PD lease |
| **Log Module** | pppoe |
| **Log Group** | dhcpv6 |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_FSM_ACTION_FAILED |
| **Message** | %s session for subscriber-id %s, MAC %s, S-Tag %s, C-Tag %s action %s failed |
| **Description** | PPP FSM action failed |
| **Log Module** | pppoe |
| **Log Group** | ppp |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PPPOE_PADT_RECEIVED |
| **Message** | PPPoE PADT received for subscriber-id %s on interface %s, MAC %s, S-Tag %s, C-Tag %s |
| **Description** | Received PPPoE discovery packet with matching configuration |
| **Log Module** | pppoe |
| **Log Group** | config |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

## Routing Information Base (RIB) Log Messages

### Log Level Summary

Info(7)

| | |
|---|---|
| **Log ID** | RIB_OPER_ROUTER_ID_ADD |
| **Message** | Add router-id %s, instance %s |
| **Description** | Router-ID configuration change triggered |
| **Log Module** | rib |
| **Log Group** | config |
| **Severity** | info |
| **Recommended Actions** | Check the router-id configurations for this instance |

| | |
|---|---|
| **Log ID** | RIB_OPER_ROUTER_ID_DEL |
| **Message** | Delete router-id %s, instance %s |
| **Description** | Router-ID configuration change triggered |
| **Log Module** | rib |
| **Log Group** | config |
| **Severity** | info |
| **Recommended Actions** | Check the router-id configurations for this instance |

| | |
|---|---|
| **Log ID** | RIB_OPER_ROUTER_ID_UPDATE |
| **Message** | Change router-id, instance %s, from %s to %s |
| **Description** | Router-ID configuration change triggered |
| **Log Module** | rib |
| **Log Group** | config |
| **Severity** | info |
| **Recommended Actions** | Check the router-id configurations for this instance |

| | |
|---|---|
| **Log ID** | RIB_OPER_IPV4_NEIGHBOR_ADD |
| **Message** | Add ipv4 neighbor, interface %s, address %s |
| **Description** | New ipv4 neighbor is detected |
| **Log Module** | rib |
| **Log Group** | neighbor-events |
| **Severity** | info |
| **Recommended Actions** | Check the ipv4 neighbor details |

| | |
|---|---|
| **Log ID** | RIB_OPER_IPV4_NEIGHBOR_DELETE |
| **Message** | Delete ipv4 neighbor, interface %s, address %s |
| **Description** | The ipv4 neighbor is deleted |
| **Log Module** | rib |
| **Log Group** | neighbor-events |
| **Severity** | info |
| **Recommended Actions** | Check the ifl status |

| | |
|---|---|
| **Log ID** | RIB_OPER_IPV6_NEIGHBOR_ADD |
| **Message** | Add ipv6 neighbor, interface %s, address %s |
| **Description** | New ipv6 neighbor is detected |
| **Log Module** | rib |
| **Log Group** | neighbor-events |
| **Severity** | info |

| | |
|---|---|
| **Recommended Actions** | Check the ipv6 neighbor details |

| | |
|---|---|
| **Log ID** | RIB_OPER_IPV6_NEIGHBOR_DELETE |
| **Message** | Delete ipv6 neighbor, interface %s, address %s |
| **Description** | The ipv6 neighbor is deleted |
| **Log Module** | rib |
| **Log Group** | neighbor-events |
| **Severity** | info |
| **Recommended Actions** | Check the ipv6 neighbor details |

## Subscriber Management Log Messages

### Log Level Summary

Error(10), Warning(9), Notice(24), Info(15), Debug(43)

| | |
|---|---|
| **Log ID** | SUBSMGMT_TRAP_DEL |
| **Message** | Failed to delete I/O trap %s |
| **Description** | Failed to delete I/O trap entry |
| **Log Module** | subsMgmt |
| **Log Group** | subscriber |
| **Severity** | error |
| **Recommended Actions** | Check FIB/RIB logs |

| | |
|---|---|
| **Log ID** | SUBSMGMT_BDS_TABLE_CREATE_ERROR |
| **Message** | Failed to create BDS table %s |
| **Description** | Failed to create BDS table |
| **Log Module** | subsMgmt |
| **Log Group** | bds |
| **Severity** | error |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_BDS_TEMPLATE_GET_ERROR |
| **Message** | Failed to get BDS object template %s |
| **Description** | Failed to get BDS object template |
| **Log Module** | subsMgmt |
| **Log Group** | bds |
| **Severity** | error |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_BDS_TEMPLATE_POPULATE_ERROR |
| **Message** | Failed to populate BDS object template %s |
| **Description** | Failed to populate BDS object template |
| **Log Module** | subsMgmt |
| **Log Group** | bds |
| **Severity** | error |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_BDS_OBJECT_ADD_ERROR |
| **Message** | Failed to add BDS object to table %s |
| **Description** | Failed to add BDS object to table |
| **Log Module** | subsMgmt |
| **Log Group** | bds |
| **Severity** | error |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_BDS_OBJECT_DEL_ERROR |
| **Message** | Failed to delete BDS object from table %s |
| **Description** | Failed to delete BDS object from table |
| **Log Module** | subsMgmt |

| | |
|---|---|
| **Log Group** | bds |
| **Severity** | error |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RADIUS_SERVER_DELETE_FAILED |
| **Message** | Failed to delete RADIUS server %s |
| **Description** | RADIUS server delete failed |
| **Log Module** | subsMgmt |
| **Log Group** | radius |
| **Severity** | error |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RADIUS_AUTH_REQ_ERROR |
| **Message** | Failed to add RADIUS authentication request for subscriber %s from server %s |
| **Description** | RADIUS authentication request failure |
| **Log Module** | subsMgmt |
| **Log Group** | radius |
| **Severity** | error |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RADIUS_ACCOUNTING_REGUEST_ERROR |
| **Message** | Failed to add RADIUS accounting request for subscriber %s from server %s |
| **Description** | RADIUS accounting request failure |
| **Log Module** | subsMgmt |
| **Log Group** | radius |
| **Severity** | error |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_L2BSA_GENERIC_ERROR |
| **Message** | %s |
| **Description** | General subscriber L2BSA related errors |
| **Log Module** | subsMgmt |
| **Log Group** | subscriber |
| **Severity** | error |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_AAA_PROFILE_NOT_FOUND |
| **Message** | AAA configuration profile %s not found for subscriber %s |
| **Description** | AAA configuration profile subscriber events |
| **Log Module** | subsMgmt |
| **Log Group** | config |
| **Severity** | warning |
| **Recommended Actions** | Check the AAA profile configuration |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RADIUS_PROFILE_MISSING_IN_AAA |
| **Message** | RADIUS profile missing in AAA profile %s for subscriber %s |
| **Description** | Missing RADIUS profile in AAA profile |
| **Log Module** | subsMgmt |
| **Log Group** | config |
| **Severity** | warning |
| **Recommended Actions** | Check the RADIUS profile in AAA profile configuration |

| | |
|---|---|
| **Log ID** | SUBSMGMT_MISSING_AAA_CONFIG |
| **Message** | missing AAA %s for subscriber %s |
| **Description** | Subscriber AAA configuration warning |
| **Log Module** | subsMgmt |

| | |
|---|---|
| **Log Group** | config |
| **Severity** | warning |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_INTERNAL_AUTHENTICATION |
| **Message** | Failed to %s during authentication for subscriber %s |
| **Description** | General subscriber authentication warnings |
| **Log Module** | subsMgmt |
| **Log Group** | subscriber |
| **Severity** | warning |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_INTERNAL_ACCOUNTING |
| **Message** | Failed to %s during accounting for subscriber %s |
| **Description** | General subscriber accounting warnings |
| **Log Module** | subsMgmt |
| **Log Group** | subscriber |
| **Severity** | warning |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_IFL_DELETE |
| **Message** | Failed to delete %s for ifl %s |
| **Description** | Subscriber logical interface delete events |
| **Log Module** | subsMgmt |
| **Log Group** | subscriber |
| **Severity** | warning |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_BDS_TABLE_GET_ERROR |
| **Message** | Failed to get BDS table %s |
| **Description** | Failed to get BDS table / not found |
| **Log Module** | subsMgmt |
| **Log Group** | bds |
| **Severity** | warning |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RADIUS_ACCOUNTING_QUEUE_INDEX |
| **Message** | RADIUS accounting queue index exhausted |
| **Description** | RADIUS accounting queue index |
| **Log Module** | subsMgmt |
| **Log Group** | radius |
| **Severity** | warning |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RADIUS_AUTHENTICATION_QUEUE_INDEX |
| **Message** | RADIUS authentication queue index exhausted |
| **Description** | RADIUS authentication queue index |
| **Log Module** | subsMgmt |
| **Log Group** | radius |
| **Severity** | warning |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_TRAP_ADD |
| **Message** | Failed to add I/O trap %s |
| **Description** | Failed to add I/O trap entry |
| **Log Module** | subsMgmt |
| **Log Group** | subscriber |
| **Severity** | notice |

| | |
|---|---|
| **Recommended Actions** | Check FIB/RIB logs |

| | |
|---|---|
| **Log ID** | SUBSMGMT_QOS_SHAPER |
| **Message** | Invalid dynamic shaper definition %s for subscriber %s |
| **Description** | Invalid subscriber QoS shaper definition received |
| **Log Module** | subsMgmt |
| **Log Group** | subscriber |
| **Severity** | notice |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_QOS_POLICER |
| **Message** | Invalid dynamic policer definition %s for subscriber %s |
| **Description** | Invalid subscriber QoS policer definition received |
| **Log Module** | subsMgmt |
| **Log Group** | subscriber |
| **Severity** | notice |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_IPV4_DUPLICATE_ADDRESS_DETECTED |
| **Message** | Duplicate IPv4 address %s detected for subscriber %s with subscriber %s |
| **Description** | IPv4 Duplicate address detected |
| **Log Module** | subsMgmt |
| **Log Group** | subscriber |
| **Severity** | notice |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_IPV6_DUPLICATE_ADDRESS_DETECTED |
| **Message** | Duplicate IPv6 prefix %s detected for subscriber %s with subscriber %s |

| | |
|---|---|
| **Description** | IPv6 Duplicate address detected |
| **Log Module** | subsMgmt |
| **Log Group** | subscriber |
| **Severity** | notice |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_INSTANCE_NOT_FOUND |
| **Message** | Instance %s not found for subscriber %s |
| **Description** | Instance not found |
| **Log Module** | subsMgmt |
| **Log Group** | subscriber |
| **Severity** | notice |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_POOL_IPV4_EXHAUSTED |
| **Message** | IPv4 pool %s exhausted for subscriber %s |
| **Description** | Local IPv4 address pool exhausted |
| **Log Module** | subsMgmt |
| **Log Group** | pool |
| **Severity** | notice |
| **Recommended Actions** | Free up IPv4 addresses |

| | |
|---|---|
| **Log ID** | SUBSMGMT_POOL_IPV6_EXHAUSTED |
| **Message** | IPv6 pool %s exhausted for subscriber %s |
| **Description** | Local IPv6 address pool exhausted |
| **Log Module** | subsMgmt |
| **Log Group** | pool |
| **Severity** | notice |
| **Recommended Actions** | Free up IPv6 addresses |

| | |
|---|---|
| **Log ID** | SUBSMGMT_QOS_QUEUE |
| **Message** | Invalid dynamic queue definition %s for subscriber %s |
| **Description** | Invalid subscriber QoS queue definition received |
| **Log Module** | subsMgmt |
| **Log Group** | subscriber |
| **Severity** | notice |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_ACL_VALIDATION |
| **Message** | ACL %s attachment to subscriber %s failed, %s |
| **Description** | Subscriber ACL validation failed |
| **Log Module** | subsMgmt |
| **Log Group** | radius |
| **Severity** | notice |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_IFL_CREATE |
| **Message** | Failed to create %s for subscriber %s |
| **Description** | Subscriber logical interface errors |
| **Log Module** | subsMgmt |
| **Log Group** | subscriber |
| **Severity** | notice |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RADIUS_ACCOUNTING_EVENT |
| **Message** | Failed to %s RADIUS accounting request from %s |
| **Description** | RADIUS accounting events |
| **Log Module** | subsMgmt |

| | |
|---|---|
| **Log Group** | radius |
| **Severity** | notice |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RADIUS_TUNNEL_ERROR |
| **Message** | Tunnel error %s for subscriber %s |
| **Description** | RADIUS subscriber tunnel releated errors |
| **Log Module** | subsMgmt |
| **Log Group** | radius |
| **Severity** | notice |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RADIUS_AUTHENTICATION_ERROR |
| **Message** | Authentication error: %s for subscriber %s |
| **Description** | RADIUS authentication releated errors |
| **Log Module** | subsMgmt |
| **Log Group** | radius |
| **Severity** | notice |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RADIUS_ACCOUNTING_ERROR |
| **Message** | Accounting error: %s for subscriber %s |
| **Description** | RADIUS accounting releated errors |
| **Log Module** | subsMgmt |
| **Log Group** | radius |
| **Severity** | notice |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RADIUS_ENCODE_ERROR |
| **Message** | Encode error %s for subscriber %s |
| **Description** | RADIUS encode errors |
| **Log Module** | subsMgmt |
| **Log Group** | radius |
| **Severity** | notice |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RADIUS_ASCEND_DATA_FILTER |
| **Message** | Ascend-Data-Filter error for subscriber %s due to %s |
| **Description** | RADIUS filter message decode error |
| **Log Module** | subsMgmt |
| **Log Group** | radius |
| **Severity** | notice |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RADIUS_ASCEND_DATA_FILTER_NOT_SUPPORTED |
| **Message** | Ascend-Data-Filter attribute %s with value %x not supported for subscriber %s |
| **Description** | RADIUS filter message decode error |
| **Log Module** | subsMgmt |
| **Log Group** | radius |
| **Severity** | notice |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RD_INVALID_OBJECT |
| **Message** | Redundancy object with invalid subscriber-id/rd-session-id |
| **Description** | Redundancy invalid object |
| **Log Module** | subsMgmt |
| **Log Group** | redundancy |

| | |
|---|---|
| **Severity** | notice |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RD_INVALID_EVENT |
| **Message** | Invalid event on redundancy session %s, %s |
| **Description** | Redundancy invalid event |
| **Log Module** | subsMgmt |
| **Log Group** | redundancy |
| **Severity** | notice |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RD_ADD_FAILED |
| **Message** | Failed to add subscriber %s for redundancy session %s |
| **Description** | Redundancy sync failed |
| **Log Module** | subsMgmt |
| **Log Group** | redundancy |
| **Severity** | notice |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RD_OBJECT_VALIDATION_FAILED |
| **Message** | Object validation failed for subscriber %s in redundancy session %s due to %s |
| **Description** | Redundancy invalid object |
| **Log Module** | subsMgmt |
| **Log Group** | redundancy |
| **Severity** | notice |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_FRAMED_ROUTE |
| **Message** | Framed-Route error for subscriber %s due to %s |
| **Description** | RADIUS Framed-Route handling error |
| **Log Module** | subsMgmt |
| **Log Group** | radius |
| **Severity** | notice |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_FRAMED_IPV6_ROUTE |
| **Message** | Framed-IPv6-Route error for subscriber %s due to %s |
| **Description** | RADIUS Framed-IPv6-Route handling error |
| **Log Module** | subsMgmt |
| **Log Group** | radius |
| **Severity** | notice |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_AAA_PROFILE_INVALID_ORDER |
| **Message** | AAA configuration profile %s invalid configuration order for subscriber %s |
| **Description** | AAA configuration profile subscriber events |
| **Log Module** | subsMgmt |
| **Log Group** | config |
| **Severity** | info |
| **Recommended Actions** | Check the authentication/accounting order in AAA profile configuration |

| | |
|---|---|
| **Log ID** | SUBSMGMT_AUTHENTICATION_FAILED |
| **Message** | Authentication failed for subscriber %s due to %s |
| **Description** | General subscriber authentication events |
| **Log Module** | subsMgmt |
| **Log Group** | subscriber |

| | |
|---|---|
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_ACCT_SESSION_ID_NOT_FOUND |
| **Message** | No subscriber with Acct-Session-ID %s found |
| **Description** | CoA request with unknown Acct-Session-ID received |
| **Log Module** | subsMgmt |
| **Log Group** | subscriber |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_SUBSCRIBER_ADD |
| **Message** | Subscriber %s added |
| **Description** | New subscriber added |
| **Log Module** | subsMgmt |
| **Log Group** | subscriber |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_SUBSCRIBER_DEL |
| **Message** | Subscriber %s deleted |
| **Description** | Subscriber deleted |
| **Log Module** | subsMgmt |
| **Log Group** | subscriber |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_SUBSCRIBER_NOT_FOUND |

| | |
|---|---|
| **Message** | Subscriber %s not found |
| **Description** | Subscriber not found |
| **Log Module** | subsMgmt |
| **Log Group** | subscriber |
| **Severity** | info |
| **Recommended Actions** | Different subscriber related actions |

| | |
|---|---|
| **Log ID** | SUBSMGMT_FSM_EVENT |
| **Message** | Subscriber %s FSM State change, from %s, event %s, to %s |
| **Description** | Subscriber FSM state transition |
| **Log Module** | subsMgmt |
| **Log Group** | subscriberfsm |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_SERVICE_PROFILE_GET_FAILED |
| **Message** | Failed to get service profile %s for subscriber %s |
| **Description** | Failed to get service configuration profile |
| **Log Module** | subsMgmt |
| **Log Group** | config |
| **Severity** | info |
| **Recommended Actions** | Reconfigure the service profile |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RADIUS_PROFILE_NOT_FOUND |
| **Message** | RADIUS profile %s not found |
| **Description** | RADIUS profile not found events |
| **Log Module** | subsMgmt |
| **Log Group** | radius |
| **Severity** | info |

| | |
|---|---|
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RADIUS_SERVER_NOT_FOUND |
| **Message** | RADIUS server %s not found |
| **Description** | RADIUS server not found events |
| **Log Module** | subsMgmt |
| **Log Group** | radius |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_LI_SUBSCRIBER_NOT_FOUND |
| **Message** | Subscriber %s not found, Lawful Intercept request is aborted |
| **Description** | Subscriber not found |
| **Log Module** | subsMgmt |
| **Log Group** | lawful-intercept |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_LI_IFL_NOT_FOUND |
| **Message** | Subscriber IFL not found for %s , Lawful Intercept request is aborted |
| **Description** | Subscriber Lawful Intercept IFL not found |
| **Log Module** | subsMgmt |
| **Log Group** | lawful-intercept |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_LI_INVALID_ATTRIBUTE |
| **Message** | Invalid Lawful Intercept attribute %s for subscriber %s |

| | |
|---|---|
| **Description** | Subscriber Lawful Intercept Errors |
| **Log Module** | subsMgmt |
| **Log Group** | lawful-intercept |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_LI_DECRYPTION_FAILED |
| **Message** | RADIUS Lawful Intercept request attribute %s decryption failed %s for subscriber %s |
| **Description** | Subscriber Lawful Intercept Errors |
| **Log Module** | subsMgmt |
| **Log Group** | lawful-intercept |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_LI_MISSING_SUBSCRIBER_ID |
| **Message** | Mandatory subscriber_id is missing, Lawful Intercept request is aborted |
| **Description** | Subscriber Lawful Intercept Errors |
| **Log Module** | subsMgmt |
| **Log Group** | lawful-intercept |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_FUNCTION_INVOCATION |
| **Message** | Subscriber Daemon %s Function invoked |
| **Description** | Function calls in SubscriberD |
| **Log Module** | subsMgmt |
| **Log Group** | general |
| **Severity** | debug |

| | |
|---|---|
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_PPPOE_RESULT |
| **Message** | Failed to %s from PPPoE for subscriber %s |
| **Description** | General subscriber PPPoE related errors |
| **Log Module** | subsMgmt |
| **Log Group** | subscriber |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_L2TP_RESULT |
| **Message** | Failed to extract the L2TP result |
| **Description** | General subscriber L2TP related errors |
| **Log Module** | subsMgmt |
| **Log Group** | subscriber |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_ACCOUNTING_STOP_PROGRESS |
| **Message** | Accounting STOP in progress for subscriber %s |
| **Description** | Subscriber waits for final counter update |
| **Log Module** | subsMgmt |
| **Log Group** | subscriber |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_QOS_CREATE |
| **Message** | QoS settings created for subscriber %s with IFL %s |

| | |
|---|---|
| **Description** | Add subscriber QoS settings |
| **Log Module** | subsMgmt |
| **Log Group** | subscriber |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_QOS_RESET |
| **Message** | QoS settings reset for subscriber %s with IFL %s |
| **Description** | Reset subscriber QoS settings |
| **Log Module** | subsMgmt |
| **Log Group** | subscriber |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_QOS_UPDATE |
| **Message** | QoS settings updated for subscriber %s with IFL %s |
| **Description** | Update subscriber QoS settings |
| **Log Module** | subsMgmt |
| **Log Group** | subscriber |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_BDS_TABLE_CREATED |
| **Message** | BDS table %s created |
| **Description** | BDS table created |
| **Log Module** | subsMgmt |
| **Log Group** | bds |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RADIUS_PROFILE_ACTIVATE |
| **Message** | Failed to activate RADIUS profile |
| **Description** | RADIUS profile activation event |
| **Log Module** | subsMgmt |
| **Log Group** | radius |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RADIUS_PROFILE_ADDED |
| **Message** | RADIUS Profile %s added |
| **Description** | RADIUS profile add events |
| **Log Module** | subsMgmt |
| **Log Group** | radius |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RADIUS_PROFILE_DELETED |
| **Message** | RADIUS profile %s deleted |
| **Description** | RADIUS profile delete events |
| **Log Module** | subsMgmt |
| **Log Group** | radius |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RADIUS_SERVER_ADDED |
| **Message** | RADIUS server %s added |
| **Description** | RADIUS server add events |
| **Log Module** | subsMgmt |

| | |
|---|---|
| **Log Group** | radius |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RADIUS_SERVER_DELETED |
| **Message** | RADIUS server %s deleted |
| **Description** | RADIUS server delete events |
| **Log Module** | subsMgmt |
| **Log Group** | radius |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RADIUS_SERVER_ACCOUNTING_DISABLED |
| **Message** | RADIUS server %s accounting disabled |
| **Description** | RADIUS server accounting disabled |
| **Log Module** | subsMgmt |
| **Log Group** | radius |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RADIUS_SERVER_AUTHENTICATION_DISABLED |
| **Message** | RADIUS server %s authentication disabled |
| **Description** | RADIUS server authentication disabled |
| **Log Module** | subsMgmt |
| **Log Group** | radius |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RADIUS_SERVER_UNKNOWN |
| **Message** | RADIUS message from unknown server %s |
| **Description** | RADIUS requests from unknown IPv4 addresses |
| **Log Module** | subsMgmt |
| **Log Group** | radius |
| **Severity** | debug |
| **Recommended Actions** | Check the RADIUS server configuration |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RADIUS_SERVER_GET_FAILED |
| **Message** | RADIUS server %s get failed while decoding RADIUS message %s |
| **Description** | RADIUS server get failed |
| **Log Module** | subsMgmt |
| **Log Group** | radius |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RADIUS_AUTH_REQ |
| **Message** | Added RADIUS authentication request for subscriber %s from server %s |
| **Description** | RADIUS authentication request |
| **Log Module** | subsMgmt |
| **Log Group** | radius |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RADIUS_ACCOUNTING_REQUEST |
| **Message** | Added RADIUS accounting request for subscriber %s from server %s |
| **Description** | RADIUS accounting request |
| **Log Module** | subsMgmt |
| **Log Group** | radius |
| **Severity** | debug |

| | |
|---|---|
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RADIUS_AUTH_STATE_CHANGE |
| **Message** | RADIUS Server %s Authentication State changed from %s to %s |
| **Description** | RADIUS server authentication state change |
| **Log Module** | subsMgmt |
| **Log Group** | radius |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RADIUS_ACCT_STATE_CHANGE |
| **Message** | RADIUS Server %s Accounting State changed from %s to %s |
| **Description** | RADIUS server accounting state change |
| **Log Module** | subsMgmt |
| **Log Group** | radius |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RADIUS_PROFILE_STATE_CHANGE |
| **Message** | RADIUS Profile %s State changed from %s to %s |
| **Description** | RADIUS profile accounting state change |
| **Log Module** | subsMgmt |
| **Log Group** | radius |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RADIUS_PROFILE_ERROR |
| **Message** | RADIUS profile %s %s for subscriber %s |

| | |
|---|---|
| **Description** | RADIUS profile subscriber events |
| **Log Module** | subsMgmt |
| **Log Group** | radius |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RADIUS_DROPPED |
| **Message** | RADIUS message from %s dropped due to %s |
| **Description** | RADIUS message dropped |
| **Log Module** | subsMgmt |
| **Log Group** | radius |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_FSM_ERROR |
| **Message** | FSM Error for subscriber %s, State %s, Event %s |
| **Description** | Invalid event for current state of subscriber FSM |
| **Log Module** | subsMgmt |
| **Log Group** | subscriberfsm |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_FSM_NOT_FOUND |
| **Message** | Subscriber %s not found for FSM event %s |
| **Description** | Subscriber not found for received FSM event |
| **Log Module** | subsMgmt |
| **Log Group** | subscriberfsm |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_FSM_STATIC_IP_ALLOCATION |
| **Message** | Static IPv4 address from access-profile allocated for subscriber %s |
| **Description** | Subscriber FSM event logging |
| **Log Module** | subsMgmt |
| **Log Group** | subscriberfsm |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_FSM_GENERAL_ERROR |
| **Message** | %s for subscriber %s |
| **Description** | Subscriber FSM Error logging |
| **Log Module** | subsMgmt |
| **Log Group** | subscriberfsm |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_POOL_CONFIG |
| **Message** | Failed to process pool configuration |
| **Description** | Subscriber Mgmt Pool warning |
| **Log Module** | subsMgmt |
| **Log Group** | pool |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_POOL_ASSIGN_IPV4 |
| **Message** | Failed to assign ipv4 address for subscriber %s |
| **Description** | Subscriber Mgmt Pool warning |
| **Log Module** | subsMgmt |

| | |
|---|---|
| **Log Group** | pool |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_POOL_RELEASE_IPV4 |
| **Message** | Failed to release ipv4 address for subscriber %s |
| **Description** | Subscriber Mgmt Pool warning |
| **Log Module** | subsMgmt |
| **Log Group** | pool |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_POOL_ASSIGN_IPV6 |
| **Message** | Failed to assign ipv6 address for subscriber %s |
| **Description** | Subscriber Mgmt Pool warning |
| **Log Module** | subsMgmt |
| **Log Group** | pool |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_POOL_RELEASE_IPV6 |
| **Message** | Failed to release ipv6 address for subscriber %s |
| **Description** | Subscriber Mgmt Pool warning |
| **Log Module** | subsMgmt |
| **Log Group** | pool |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_POOL_IPV4_SUCCESS |
| **Message** | IPv4 address %s from pool %s assigned to subscriber %s |
| **Description** | IPv4 address from local address pool assigned |
| **Log Module** | subsMgmt |
| **Log Group** | pool |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_POOL_IPV6_SUCCESS |
| **Message** | IPv6 prefix %s from pool %s assigned to subscriber %s |
| **Description** | IPv6 prefix from local address pool assigned |
| **Log Module** | subsMgmt |
| **Log Group** | pool |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_TEST_AAA_RESULT |
| **Message** | Failed to %s from test aaa request for subscriber %s |
| **Description** | General test aaa request subscriber related errors |
| **Log Module** | subsMgmt |
| **Log Group** | subscriber |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_IPOE_RESULT |
| **Message** | Failed to %s from IPoE for subscriber %s |
| **Description** | General subscriber IPoE related errors |
| **Log Module** | subsMgmt |
| **Log Group** | subscriber |
| **Severity** | debug |

| | |
|---|---|
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_L2BSA_RESULT |
| **Message** | Failed to %s from L2BSA service for subscriber %s |
| **Description** | General L2BSA subscriber related errors |
| **Log Module** | subsMgmt |
| **Log Group** | l2bsa |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_PLATFORM_TYPE |
| **Message** | Platform type set to %s |
| **Description** | Platform type updated |
| **Log Module** | subsMgmt |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_RD_SUBSCRIBER_EVENT |
| **Message** | Event for subscriber %s on redundancy session %s %s |
| **Description** | Redundancy subscriber event |
| **Log Module** | subsMgmt |
| **Log Group** | redundancy |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_LI_REQUEST_FAILURE |
| **Message** | Failed to process the Lawful Intercept request for subscriber %s |

| | |
|---|---|
| **Description** | Subscriber Lawful Intercept Errors |
| **Log Module** | subsMgmt |
| **Log Group** | lawful-intercept |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_LI_BDS_OBJECT_ADD_ERROR |
| **Message** | Failed to add BDS object to table %s |
| **Description** | Failed to add BDS object to table |
| **Log Module** | subsMgmt |
| **Log Group** | lawful-intercept |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | SUBSMGMT_LI_BDS_OBJECT_DEL_ERROR |
| **Message** | Failed to delete BDS object from table %s |
| **Description** | Failed to delete BDS object from table |
| **Log Module** | subsMgmt |
| **Log Group** | lawful-intercept |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

## access_service Log Messages

### Log Level Summary

Error(10), Debug(3)

| | |
|---|---|
| **Log ID** | HTTPRD_ERROR_SUBSCRIBER_IFL_MISSING |
| **Message** | HTTPRD Subscriber IFL Missing: src=%s:%s, dst=%s:%s |
| **Description** | Subscriber IFL is missing in TCP context |
| **Log Module** | access_service |

| | |
|---|---|
| **Log Group** | general |
| **Severity** | error |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | HTTPRD_ERROR_SERVER_INIT_FAILED |
| **Message** | LWIP server initialization failed for instance %s |
| **Description** | LWIP server initialization failed |
| **Log Module** | access_service |
| **Log Group** | general |
| **Severity** | error |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | HTTPRD_ERROR_OBJECT_ADD |
| **Message** | Failed to add object to %s |
| **Description** | HTTPRD failed to add object |
| **Log Module** | access_service |
| **Log Group** | general |
| **Severity** | error |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | HTTPRD_ERROR_OBJECT_DEL |
| **Message** | Failed to delete object from %s |
| **Description** | HTTPRD failed to delete object |
| **Log Module** | access_service |
| **Log Group** | general |
| **Severity** | error |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | HTTPRD_ERROR_OBJECT_GET |
| **Message** | Failed to get object from %s |
| **Description** | HTTPRD failed to get object |
| **Log Module** | access_service |
| **Log Group** | general |
| **Severity** | error |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | HTTPRD_ERROR_TABLE_CREATE |
| **Message** | Failed to create table %s |
| **Description** | HTTPRD failed to create table |
| **Log Module** | access_service |
| **Log Group** | general |
| **Severity** | error |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | HTTPRD_ERROR_OBJECT_TEMPLATE |
| **Message** | Failed to get object template %s |
| **Description** | HTTPRD failed to get object template |
| **Log Module** | access_service |
| **Log Group** | general |
| **Severity** | error |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | HTTPRD_ERROR_OBJECT_TEMPLATE_POPULATE |
| **Message** | Failed to populate object template %s |
| **Description** | HTTPRD failed to populate object template |
| **Log Module** | access_service |
| **Log Group** | general |
| **Severity** | error |

| | |
|---|---|
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | HTTPRD_ERROR_OBJECT_EXTRACT |
| **Message** | Failed to extract object template %s |
| **Description** | HTTPRD failed to extract object template |
| **Log Module** | access_service |
| **Log Group** | general |
| **Severity** | error |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | HTTPRD_ERROR_TABLE_GET |
| **Message** | Failed to get table %s |
| **Description** | HTTPRD failed to get table |
| **Log Module** | access_service |
| **Log Group** | general |
| **Severity** | error |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | HTTPRD_TCP_INFO |
| **Message** | HTTPRD TCP %s: ifl=%s, src=%s:%s, dst=%s:%s |
| **Description** | HTTPRD TCP information |
| **Log Module** | access_service |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | HTTPRD_REDIRECTED |
| **Message** | HTTPRD Redirected: ifl=%s, src=%s:%s, dst=%s:%s, url=%s |

| | |
|---|---|
| **Description** | HTTPRD redirected (302) |
| **Log Module** | access_service |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | HTTPRD_NOT_FOUND |
| **Message** | HTTPRD Not Found: ifl=%s, src=%s:%s, dst=%s:%s |
| **Description** | HTTPRD redirect information not found (404) |
| **Log Module** | access_service |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

### L2TP Log Messages

#### Log Level Summary

Notice(1), Info(14), Debug(15)

| | |
|---|---|
| **Log ID** | L2TP_TRAP_ADD |
| **Message** | Failed to add I/O trap %s |
| **Description** | Failed to add I/O trap entry |
| **Log Module** | l2tp |
| **Log Group** | config |
| **Severity** | notice |
| **Recommended Actions** | Check FIB/RIB logs |

| | |
|---|---|
| **Log ID** | L2TP_SESSION_CREATED |
| **Message** | L2TP session created with tunnel %s to %s, session %s for subscriber %s |
| **Description** | L2TP session created |
| **Log Module** | l2tp |

| | |
|---|---|
| **Log Group** | subscriber |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | L2TP_SESSION_FSM_STATE_CHANGE |
| **Message** | L2TP session %s subscriber %s, FSM state change from %s to %s for input %s with action %s |
| **Description** | L2TP session FSM state changed |
| **Log Module** | l2tp |
| **Log Group** | subscriber |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | L2TP_SESSION_REQUEST_ADD |
| **Message** | L2TP session request added for subscriber %s tunnel count %s |
| **Description** | L2TP session request added |
| **Log Module** | l2tp |
| **Log Group** | subscriber |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | L2TP_TUNNEL_REQUEST_ADD |
| **Message** | L2TP tunnel request added for tunnel %s to %s |
| **Description** | L2TP tunnel request added |
| **Log Module** | l2tp |
| **Log Group** | tunnel |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | L2TP_TUNNEL_FSM_STATE_CHANGE |
| **Message** | L2TP tunnel %s to %s, FSM state change from %s to %s for input %s with action %s |
| **Description** | L2TP tunnel FSM state changed |
| **Log Module** | l2tp |
| **Log Group** | tunnel |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | L2TP_AVP_ENCODE_ERROR |
| **Message** | L2TP AVP encode error for tunnel %s, session %s and type %s |
| **Description** | L2TP AVP encode error |
| **Log Module** | l2tp |
| **Log Group** | tunnel |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | L2TP_PACKET_ENCODE_ERROR |
| **Message** | L2TP tunnel encode error for tunnel %s to %s |
| **Description** | L2TP packet encode error |
| **Log Module** | l2tp |
| **Log Group** | tunnel |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | L2TP_TUNNEL_DECODE_ERROR |
| **Message** | L2TP tunnel decode error %s with packet length %s for tunnel %s to %s |
| **Description** | L2TP tunnel decode error |
| **Log Module** | l2tp |
| **Log Group** | tunnel |

| | |
|---|---|
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | L2TP_TUNNEL_AVP_MH_MISMATCH |
| **Message** | L2TP tunnel AVP with vendor-id %s and attribute-type %s m/h bit mismatch for tunnel %s to %s |
| **Description** | L2TP tunnel decode error |
| **Log Module** | l2tp |
| **Log Group** | tunnel |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | L2TP_PACKET_DROP |
| **Message** | L2TP Packet dropped due to %s packet type %s received ns %s tunnel nr %s on tunnel %s to %s and session %s |
| **Description** | L2TP tunnel packet dropped because duplicate or out of order |
| **Log Module** | l2tp |
| **Log Group** | tunnel |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | L2TP_MBIT_UNKNOWN |
| **Message** | L2TP tunnel unknown mandatory AVP vendor-id %s and attribute-type %s received for tunnel %s to %s |
| **Description** | L2TP tunnel packet received with unknown mandatory AVP |
| **Log Module** | l2tp |
| **Log Group** | tunnel |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | L2TP_AVP_DECODE_ERROR |
| **Message** | L2TP AVP decode error %s for vendor %s attribute-type %s for tunnel %s to %s |
| **Description** | L2TP AVP decode error |
| **Log Module** | l2tp |
| **Log Group** | tunnel |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | L2TP_TUNNEL_AUTH_ERROR |
| **Message** | L2TP tunnel %s to %s authentication error %s |
| **Description** | L2TP tunnel authentication error |
| **Log Module** | l2tp |
| **Log Group** | tunnel |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | L2TP_TUNNEL_DELETED |
| **Message** | L2TP tunnel %s to %s deleted |
| **Description** | L2TP tunnel deleted |
| **Log Module** | l2tp |
| **Log Group** | tunnel |
| **Severity** | info |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | L2TP_PACKET_RECEIVE_ERROR |
| **Message** | Failed to receive L2TP packet from SHM |
| **Description** | L2TP packet receive error |
| **Log Module** | l2tp |

| | |
|---|---|
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | L2TP_INVALID_PACKET_RECEIVED |
| **Message** | L2TP invalid packet %s received from %s |
| **Description** | L2TP invalid packet received |
| **Log Module** | l2tp |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | L2TP_SUBSCRIBER_RESULT_ERROR |
| **Message** | L2TP subcriberd result not found for tunnel %s with session %s for subscriber %s |
| **Description** | Generic L2TP subscriber result error |
| **Log Module** | l2tp |
| **Log Group** | subscriber |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | L2TP_RADIUS_TUNNEL |
| **Message** | L2TP error in RADIUS tunnel definition for subscriber %s |
| **Description** | L2TP error in RADIUS tunnel definition |
| **Log Module** | l2tp |
| **Log Group** | subscriber |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | L2TP_NO_TUNNEL_POOL |
| **Message** | L2TP no tunnel pool for subscriber %s |
| **Description** | L2TP no tunnel pool for subscriber |
| **Log Module** | l2tp |
| **Log Group** | subscriber |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | L2TP_POOL_NOT_FOUND |
| **Message** | L2TP pool %s not found for subscriber %s |
| **Description** | L2TP pool error for subscriber |
| **Log Module** | l2tp |
| **Log Group** | subscriber |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | L2TP_TUNNEL_CREDENTIAL |
| **Message** | L2TP tunnel with missing %s for subscriber %s |
| **Description** | L2TP tunnel with missing credentials |
| **Log Module** | l2tp |
| **Log Group** | subscriber |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | L2TP_TUNNEL_SEND |
| **Message** | L2TP tunnel sent for tunnel %s type %s |
| **Description** | L2TP tunnel packet send |
| **Log Module** | l2tp |
| **Log Group** | tunnel |
| **Severity** | debug |

| | |
|---|---|
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | L2TP_TX_QUEUE_ADDED |
| **Message** | L2TP tunnel %s packet %s added to TX queue |
| **Description** | L2TP packet added to tunnel TX queue |
| **Log Module** | l2tp |
| **Log Group** | tunnel |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | L2TP_TX_QUEUE_ADD_FAILED |
| **Message** | L2TP tunnel %s failed to add packet to TX queue |
| **Description** | Failed to add L2TP packet to tunnel TX queue |
| **Log Module** | l2tp |
| **Log Group** | tunnel |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | L2TP_TUNNEL_PACKET_SEND |
| **Message** | L2TP tunnel %s trying to send outstanding type %s ns %s nr %s pnr %s cw %s |
| **Description** | L2TP tunnel packet send |
| **Log Module** | l2tp |
| **Log Group** | tunnel |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | L2TP_TUNNEL_PACKET_RECEIVED |

| | |
|---|---|
| **Message** | L2TP %s received for tunnel %s |
| **Description** | L2TP tunnel packet received |
| **Log Module** | l2tp |
| **Log Group** | tunnel |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | L2TP_TUNNEL_MAX_RETRANSMISSION |
| **Message** | L2TP tunnel %s exceeds transmission |
| **Description** | L2TP tunnel exceeds transmission |
| **Log Module** | l2tp |
| **Log Group** | tunnel |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | L2TP_OUT_OF_ORDER_PACKET |
| **Message** | L2TP out of order packet accepted for tunnel %s |
| **Description** | L2TP out of order packet |
| **Log Module** | l2tp |
| **Log Group** | tunnel |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | L2TP_SESSION_PACKET_RECEIVED |
| **Message** | L2TP %s received for tunnel %s session %s |
| **Description** | L2TP session packet received |
| **Log Module** | l2tp |
| **Log Group** | session |
| **Severity** | debug |

| | |
|---|---|
| **Recommended Actions** | Contact Rtbrick customer support |

## IGMP Log Messages

### Log Level Summary

Info(7), Debug(12)

| | |
|---|---|
| **Log ID** | IGMP_INSTANCE_UP |
| **Message** | IGMP instance is UP - {instance:%s} |
| **Description** | IGMP instance configuration add triggered |
| **Log Module** | igmp |
| **Log Group** | config |
| **Severity** | INFO |
| **Recommended Actions** | Check the IGMP status for this instance |

| | |
|---|---|
| **Log ID** | IGMP_INSTANCE_DOWN |
| **Message** | IGMP instance is DOWN - {instance:%s} |
| **Description** | IGMP instance configuration delete triggered |
| **Log Module** | igmp |
| **Log Group** | config |
| **Severity** | INFO |
| **Recommended Actions** | Check the IGMP status for this instance |

| | |
|---|---|
| **Log ID** | IGMP_INTERFACE_UP |
| **Message** | IGMP logical interface is UP - {instance:%s logical-interface:%s} |
| **Description** | IGMP logical interface configuration add or dynamic subscriber ifl creation triggered |
| **Log Module** | igmp |
| **Log Group** | config |
| **Severity** | INFO |

| | |
|---|---|
| **Recommended Actions** | Check the IGMP status for this logical interface |

| | |
|---|---|
| **Log ID** | IGMP_INTERFACE_DOWN |
| **Message** | IGMP logical interface is DOWN - {instance:%s interface:%s} |
| **Description** | IGMP logical interface configuration delete or dynamic subscriber ifl deletion triggered |
| **Log Module** | igmp |
| **Log Group** | config |
| **Severity** | INFO |
| **Recommended Actions** | Check the IGMP status for this logical interface |

| | |
|---|---|
| **Log ID** | IGMP_INTERFACE_ROLE_QUERIER |
| **Message** | IGMP interface role is querier - {instance:%s interface:%s} |
| **Description** | IGMP interface role is querier |
| **Log Module** | igmp |
| **Log Group** | igmp-interface-events |
| **Severity** | INFO |
| **Recommended Actions** | Check the IGMP role on this logical interface |

| | |
|---|---|
| **Log ID** | IGMP_INTERFACE_ROLE_NON_QUERIER |
| **Message** | IGMP interface role is non-querier - {instance:%s interface:%s} |
| **Description** | IGMP interface role is non-querier |
| **Log Module** | igmp |
| **Log Group** | igmp-interface-events |
| **Severity** | INFO |
| **Recommended Actions** | Check the IGMP role on this logical interface |

| | |
|---|---|
| **Log ID** | IGMP_INTERFACE_MAX_LIMIT_REACHED |

| | |
|---|---|
| **Message** | IGMP membership report dropped due to max-limit reached on the interface - {instance:%s interface:%s} {source:%s group:%s} |
| **Description** | IGMP membership report dropped due to max-limit reached on the interface |
| **Log Module** | igmp |
| **Log Group** | igmp-packet-events |
| **Severity** | INFO |
| **Recommended Actions** | Check if maximum members limit on this interface |

| | |
|---|---|
| **Log ID** | IGMP_INTERFACE_FILTER_MEMBERSHIP_PERMITTED |
| **Message** | IGMP membership report is permitted - {instance:%s interface:%s} {source:%s group:%s} |
| **Description** | IGMP membership report is permitted |
| **Log Module** | igmp |
| **Log Group** | igmp-packet-events |
| **Severity** | DEBUG |
| **Recommended Actions** | Check if membership should be permitted |

| | |
|---|---|
| **Log ID** | IGMP_INTERFACE_FILTER_MEMBERSHIP_DENIED |
| **Message** | IGMP membership report is denied by interface filter - {instance:%s interface:%s} {source:%s group:%s} |
| **Description** | IGMP membership report is denied |
| **Log Module** | igmp |
| **Log Group** | igmp-packet-events |
| **Severity** | DEBUG |
| **Recommended Actions** | Check if membership report should be denied |

| | |
|---|---|
| **Log ID** | IGMP_INTERFACE_GENERAL_QUERY_SENT |
| **Message** | IGMP general query sent on the interface - {instance:%s interface:%s} |
| **Description** | IGMP general query sent on the interface |
| **Log Module** | igmp |

| | |
|---|---|
| **Log Group** | igmp-packet-events |
| **Severity** | DEBUG |
| **Recommended Actions** | Check if general query out statistics on the interface |

| | |
|---|---|
| **Log ID** | IGMP_INTERFACE_GENERAL_QUERY_RECEIVED |
| **Message** | IGMP general query received on the interface - {instance:%s interface:%s} |
| **Description** | IGMP general query received on the interface |
| **Log Module** | igmp |
| **Log Group** | igmp-packet-events |
| **Severity** | DEBUG |
| **Recommended Actions** | Check if general query in statistics on the interface |

| | |
|---|---|
| **Log ID** | IGMP_INTERFACE_GROUP_QUERY_SENT |
| **Message** | IGMP group query sent on the interface - {instance:%s interface:%s} {group:%s} |
| **Description** | IGMP group query sent on the interface |
| **Log Module** | igmp |
| **Log Group** | igmp-packet-events |
| **Severity** | DEBUG |
| **Recommended Actions** | Check if group query out statistics on the interface |

| | |
|---|---|
| **Log ID** | IGMP_INTERFACE_GROUP_QUERY_RECEIVED |
| **Message** | IGMP group query received on the interface - {instance:%s interface:%s} {group:%s} |
| **Description** | IGMP group query received on the interface |
| **Log Module** | igmp |
| **Log Group** | igmp-packet-events |
| **Severity** | DEBUG |
| **Recommended Actions** | Check if group query in statistics on the interface |

| | |
|---|---|
| **Log ID** | IGMP_INTERFACE_MEMBERSHIP_JOIN_RECEIVED |
| **Message** | IGMP membership join received on the interface - {instance:%s interface:%s} {source:%s group:%s} |
| **Description** | IGMP membership join received on the interface |
| **Log Module** | igmp |
| **Log Group** | igmp-packet-events |
| **Severity** | DEBUG |
| **Recommended Actions** | Check if IGMP membership join received on the interface |

| | |
|---|---|
| **Log ID** | IGMP_INTERFACE_MEMBERSHIP_LEAVE_RECEIVED |
| **Message** | IGMP membership leave received on the interface - {instance:%s interface:%s} {source:%s group:%s} |
| **Description** | IGMP membership leave received on the interface |
| **Log Module** | igmp |
| **Log Group** | igmp-packet-events |
| **Severity** | DEBUG |
| **Recommended Actions** | Check if IGMP membership leave received on the interface |

| | |
|---|---|
| **Log ID** | IGMP_MEMBERSHIP_ADD |
| **Message** | IGMP membership is added - {instance:%s interface:%s} {source:%s group:%s type:%s} |
| **Description** | IGMP membership is added |
| **Log Module** | igmp |
| **Log Group** | igmp-membership-events |
| **Severity** | DEBUG |
| **Recommended Actions** | Check if membership should be permitted |

| | |
|---|---|
| **Log ID** | IGMP_MEMBERSHIP_DELETE |

| | |
|---|---|
| **Message** | IGMP membership is deleted - {instance:%s interface:%s} {source:%s group:%s type:%s} |
| **Description** | IGMP membership is deleted |
| **Log Module** | igmp |
| **Log Group** | igmp-membership-events |
| **Severity** | DEBUG |
| **Recommended Actions** | Check if IGMP leave received or static join removed |

| | |
|---|---|
| **Log ID** | IGMP_FIBLOCAL_ADD |
| **Message** | IGMP mroute added to fiblocal table - {instance:%s interface:%s} {source:%s group:%s type:%s} |
| **Description** | IGMP mroute added to fiblocal table |
| **Log Module** | igmp |
| **Log Group** | igmp-route-events |
| **Severity** | DEBUG |
| **Recommended Actions** | Check if IGMP route added to fiblocal table |

| | |
|---|---|
| **Log ID** | IGMP_FIBLOCAL_DELETE |
| **Message** | IGMP mroute deleted to fiblocal table - {instance:%s interface:%s} {source:%s group:%s type:%s} |
| **Description** | IGMP mroute deleted from fiblocal table |
| **Log Module** | igmp |
| **Log Group** | igmp-route-events |
| **Severity** | DEBUG |
| **Recommended Actions** | Check if IGMP mroute deleted from fiblocal table |

## oam Log Messages

### Log Level Summary

Alert(1), Debug(17)

| | |
|---|---|
| **Log ID** | OAM_BFD_SESSION_STATUS |

| | |
|---|---|
| **Message** | OAM BFD session is %s for peer %s, local address %s |
| **Description** | BFD session status |
| **Log Module** | oam |
| **Log Group** | bfd |
| **Severity** | Alert |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | OAM_TABLE_CREATED |
| **Message** | BDS table %s created |
| **Description** | BDS table created |
| **Log Module** | oam |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | OAM_TABLE_CREATE_ERROR |
| **Message** | Failed to create BDS table %s |
| **Description** | Failed to create BDS table |
| **Log Module** | oam |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | OAM_TABLE_DELETED |
| **Message** | BDS table %s deleted |
| **Description** | BDS table deleted |
| **Log Module** | oam |
| **Log Group** | general |
| **Severity** | debug |

| | |
|---|---|
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | OAM_TABLE_ATTR_NOT_FOUND |
| **Message** | Failed to get table attribute for table %s |
| **Description** | BDS table attribute not found |
| **Log Module** | oam |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | OAM_TABLE_NOT_FOUND |
| **Message** | Failed to get table %s |
| **Description** | BDS table not found |
| **Log Module** | oam |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | OAM_TABLE_SUBSCRIBED |
| **Message** | BDS table %s subscribed |
| **Description** | BDS table subscribed |
| **Log Module** | oam |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | OAM_TABLE_UNSUBSCRIBED |
| **Message** | BDS table %s unsubscribed |

| | |
|---|---|
| **Description** | BDS table unsubscribed |
| **Log Module** | oam |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | OAM_TRAP_ADDED |
| **Message** | Trap %s added |
| **Description** | OAM trap added |
| **Log Module** | oam |
| **Log Group** | config |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | OAM_TRAP_DELETED |
| **Message** | Trap %s deleted |
| **Description** | OAM trap deleted |
| **Log Module** | oam |
| **Log Group** | config |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | OAM_OBJ_GET_FAILED |
| **Message** | Failed to get object for table %s |
| **Description** | BDS object get failed |
| **Log Module** | oam |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | OAM_OBJ_ADD_FAILED |
| **Message** | Failed to add object for table %s |
| **Description** | BDS object add failed |
| **Log Module** | oam |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | OAM_OBJ_DEL_FAILED |
| **Message** | Failed to delete object for table %s |
| **Description** | BDS object delete failed |
| **Log Module** | oam |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | OAM_CFM_SESSION_NOT_FOUND |
| **Message** | CFM session not found for ifl %s |
| **Description** | BDS object delete failed |
| **Log Module** | oam |
| **Log Group** | cfm |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | MPLS_OAM_TABLE_CREATED |
| **Message** | BDS table %s created |
| **Description** | BDS table created |
| **Log Module** | oam |

| | |
|---|---|
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | MPLS_OAM_TABLE_CREATE_ERROR |
| **Message** | Failed to create BDS table %s |
| **Description** | Failed to create BDS table |
| **Log Module** | oam |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | MPLS_OAM_TABLE_DELETED |
| **Message** | BDS table %s deleted |
| **Description** | BDS table deleted |
| **Log Module** | oam |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | MPLS_OAM_TABLE_DELETE_ERROR |
| **Message** | Failed to delete BDS table %s |
| **Description** | Failed to delete BDS table |
| **Log Module** | oam |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

## PIM Log Messages

### Log Level Summary

Error(3), Info(11), Debug(15)

| | |
|---|---|
| **Log ID** | PIM_INSTANCE_DOWN |
| **Message** | PIM Instance (%s) is Down |
| **Description** | PIM Instance Events |
| **Log Module** | pim |
| **Log Group** | instance |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PIM_NEIGHBOR_DOWN |
| **Message** | PIM Neighbor (%s) is Down on Interface (%s), Instance (%s) |
| **Description** | PIM Neighbor Events |
| **Log Module** | pim |
| **Log Group** | neighbor |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PIM_ERROR_LOG |
| **Message** | PIM Error Reason (%s) |
| **Description** | PIM Error Reason |
| **Log Module** | pim |
| **Log Group** | error-log |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PIM_INSTANCE_UP |
| **Message** | PIM Instance (%s) is Up |

| | |
|---|---|
| **Description** | PIM Instance Events |
| **Log Module** | pim |
| **Log Group** | instance |
| **Severity** | INFO |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PIM_INTERFACE_UP |
| **Message** | PIM Interface (%s) is Up on Instance (%s) |
| **Description** | PIM Interface Events |
| **Log Module** | pim |
| **Log Group** | interface |
| **Severity** | INFO |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PIM_INTERFACE_DOWN |
| **Message** | PIM Interface (%s) is Down on Instance (%s) |
| **Description** | PIM Interface Events |
| **Log Module** | pim |
| **Log Group** | interface |
| **Severity** | INFO |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PIM_NEIGHBOR_UP |
| **Message** | PIM Neighbor (%s) is Up on Interface (%s), Instance (%s) |
| **Description** | PIM Neighbor Events |
| **Log Module** | pim |
| **Log Group** | neighbor |
| **Severity** | INFO |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PIM_DR_ELECTION_WON |
| **Message** | DR election won new DR (%s) elected on Interface (%s), Instance (%s) |
| **Description** | PIM DR election Events |
| **Log Module** | pim |
| **Log Group** | dr-election |
| **Severity** | INFO |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PIM_DR_ELECTION_LOST |
| **Message** | DR election lost new DR (%s) elected on Interface (%s), Instance (%s) |
| **Description** | PIM DR election Events |
| **Log Module** | pim |
| **Log Group** | dr-election |
| **Severity** | INFO |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PIM_PACKET_SEND |
| **Message** | Sending (%s) Packet with Length %s on Instance (%s) Interface (%s) from (%s) to (%s) |
| **Description** | PIM Packet Send Events |
| **Log Module** | pim |
| **Log Group** | packet |
| **Severity** | INFO |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PIM_PACKET_RECEIVE |
| **Message** | Receiving (%s) Packet with Length %s on Instance (%s) Interface (%s) from (%s) to (%s) |
| **Description** | PIM Packet Receive Events |

| | |
|---|---|
| **Log Module** | pim |
| **Log Group** | packet |
| **Severity** | INFO |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PIM_INFO_LOG |
| **Message** | PIM Info (%s) |
| **Description** | PIM Info |
| **Log Module** | pim |
| **Log Group** | info-log |
| **Severity** | INFO |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PIM_JOIN_SENT |
| **Message** | PIM join sent for multicast route - {instance:%s interface:%s} {source:%s group:%s} |
| **Description** | PIM join sent for multicast route |
| **Log Module** | pim |
| **Log Group** | join-prune |
| **Severity** | INFO |
| **Recommended Actions** | Check if PIM join received at upstream router |

| | |
|---|---|
| **Log ID** | PIM_PRUNE_SENT |
| **Message** | PIM prune sent for multicast route - {instance:%s interface:%s} {source:%s group:%s} |
| **Description** | PIM prune sent for multicast route |
| **Log Module** | pim |
| **Log Group** | join-prune |
| **Severity** | INFO |

| | |
|---|---|
| **Recommended Actions** | Check if PIM prune received at upstream router |

| | |
|---|---|
| **Log ID** | PIM_OBJECT_TRACE |
| **Message** | Dumpping Object (%s)%s:%s %s |
| **Description** | PIM Object Operation Dump |
| **Log Module** | pim |
| **Log Group** | object-trace |
| **Severity** | DEBUG |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | PIM_RIBIN_ROUTE_ADD |
| **Message** | PIM mroute added to ribin table - {instance:%s interface:%s} {source:%s group:%s source:%s sub-source:%s} |
| **Description** | PIM mroute added to ribin table |
| **Log Module** | pim |
| **Log Group** | multicast-route |
| **Severity** | DEBUG |
| **Recommended Actions** | Check if PIM route added to ribin table |

| | |
|---|---|
| **Log ID** | PIM_RIBIN_ROUTE_CHANGE |
| **Message** | PIM mroute changed to ribin table - {instance:%s interface:%s} {source:%s group:%s source:%s sub-source:%s} |
| **Description** | PIM mroute changed to ribin table |
| **Log Module** | pim |
| **Log Group** | multicast-route |
| **Severity** | DEBUG |
| **Recommended Actions** | Check if PIM route changed to ribin table |

| | |
|---|---|
| **Log ID** | PIM_RIBIN_ROUTE_DELETE |

| | |
|---|---|
| **Message** | PIM mroute deleted to ribin table - {instance:%s interface:%s} {source:%s group:%s source:%s sub-source:%s} |
| **Description** | PIM mroute deleted to ribin table |
| **Log Module** | pim |
| **Log Group** | multicast-route |
| **Severity** | DEBUG |
| **Recommended Actions** | Check if PIM route deleted to ribin table |

| | |
|---|---|
| **Log ID** | PIM_ADDRESS_RESOLUTION_REQUEST_ADD |
| **Message** | PIM RPF address resolution requested - {instance:%s address:%s} {afi:%s safi:%s} |
| **Description** | PIM RPF address resolution requested |
| **Log Module** | pim |
| **Log Group** | rpf |
| **Severity** | DEBUG |
| **Recommended Actions** | Check requested address has resolvable route |

| | |
|---|---|
| **Log ID** | PIM_ADDRESS_RESOLUTION_REQUEST_DELETE |
| **Message** | PIM RPF address resolution request deleted - {instance:%s address:%s} {afi:%s safi:%s} |
| **Description** | PIM RPF address resolution request deleted |
| **Log Module** | pim |
| **Log Group** | rpf |
| **Severity** | DEBUG |
| **Recommended Actions** | Check if no multicast route with requested address |

| | |
|---|---|
| **Log ID** | PIM_ADDRESS_RESOLUTION_RESPONSE_ADD |
| **Message** | PIM RPF address resolution received - {instance:%s address:%s} {afi:%s safi:%s} {exit-interface:%s peer-ip:%s} |
| **Description** | PIM RPF address resolution received |
| **Log Module** | pim |

| | |
|---|---|
| **Log Group** | rpf |
| **Severity** | DEBUG |
| **Recommended Actions** | Check if resolved address in reachable |

| | |
|---|---|
| **Log ID** | PIM_ADDRESS_RESOLUTION_RESPONSE_CHANGE |
| **Message** | PIM RPF resolved address changed - {instance:%s address:%s} {afi:%s safi:%s} {exit-interface:%s peer-ip:%s} |
| **Description** | PIM RPF resolved address changed |
| **Log Module** | pim |
| **Log Group** | rpf |
| **Severity** | DEBUG |
| **Recommended Actions** | Check if resolved address in reachable |

| | |
|---|---|
| **Log ID** | PIM_ADDRESS_RESOLUTION_RESPONSE_DELETE |
| **Message** | PIM RPF resolved address deleted - {instance:%s address:%s} {afi:%s safi:%s} {exit-interface:%s peer-ip:%s} |
| **Description** | PIM RPF resolved address deleted |
| **Log Module** | pim |
| **Log Group** | rpf |
| **Severity** | DEBUG |
| **Recommended Actions** | Check if resolved address in reachable |

| | |
|---|---|
| **Log ID** | PIM_RIBLOCAL_ROUTE_ADD |
| **Message** | PIM multicast route added to riblocal table - {instance:%s interface:%s} {source:%s group:%s rpf-interface:%s} |
| **Description** | PIM multicast route added to riblocal table |
| **Log Module** | pim |
| **Log Group** | multicast-route |
| **Severity** | DEBUG |
| **Recommended Actions** | Check if installed multicast route has subscribers |

| | |
|---|---|
| **Log ID** | PIM_RIBLOCAL_ROUTE_DELETE |
| **Message** | PIM multicast route deleted to riblocal table - {instance:%s interface:%s} {source:%s group:%s rpf-interface:%s} |
| **Description** | PIM multicast route deleted to riblocal table |
| **Log Module** | pim |
| **Log Group** | multicast-route |
| **Severity** | DEBUG |
| **Recommended Actions** | Check if multicast route has no subscribers |

| | |
|---|---|
| **Log ID** | PIM_RIBLOCAL_ROUTE_CHANGE |
| **Message** | PIM multicast route changed to riblocal table - {instance:%s interface:%s} {source:%s group:%s rpf-interface:%s} |
| **Description** | PIM multicast route changed to riblocal table |
| **Log Module** | pim |
| **Log Group** | multicast-route |
| **Severity** | DEBUG |
| **Recommended Actions** | Check if multicast route has no subscribers |

| | |
|---|---|
| **Log ID** | PIM_FIBLOCAL_ROUTE_ADD |
| **Message** | PIM multicast route added to fiblocal table - {instance:%s} {source:%s group:%s rpf-interface:%s} |
| **Description** | PIM multicast route added to fiblocal table |
| **Log Module** | pim |
| **Log Group** | multicast-route |
| **Severity** | DEBUG |
| **Recommended Actions** | Check if installed multicast route has subscribers |

| | |
|---|---|
| **Log ID** | PIM_FIBLOCAL_ROUTE_DELETE |

| | |
|---|---|
| **Message** | PIM multicast route deleted to fiblocal table - {instance:%s} {source:%s group:%s rpf-interface:%s} |
| **Description** | PIM multicast route deleted to fiblocal table |
| **Log Module** | pim |
| **Log Group** | multicast-route |
| **Severity** | DEBUG |
| **Recommended Actions** | Check if multicast route has no subscribers |

| | |
|---|---|
| **Log ID** | PIM_FIBLOCAL_ROUTE_CHANGE |
| **Message** | PIM multicast route changed to fiblocal table - {instance:%s} {source:%s group:%s rpf-interface:%s} |
| **Description** | PIM multicast route changed to fiblocal table |
| **Log Module** | pim |
| **Log Group** | multicast-route |
| **Severity** | DEBUG |
| **Recommended Actions** | Check if multicast route has no subscribers |

## pool Log Messages

### Log Level Summary

Error(8)

| | |
|---|---|
| **Log ID** | POOL_CLIENT_INIT |
| **Message** | Pool client init failed for module: %s, resource: %s |
| **Description** | Pool client init failure |
| **Log Module** | pool |
| **Log Group** | internal |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | POOL_CLIENT_NOT_FOUND |

| | |
|---|---|
| **Message** | Pool client not found for module: %s, resource: %s |
| **Description** | The pool client not found |
| **Log Module** | pool |
| **Log Group** | internal |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | POOL_CLIENT_INVALID_MODULE |
| **Message** | Invalid Module name %s passed to pool client init |
| **Description** | The pool client not found |
| **Log Module** | pool |
| **Log Group** | internal |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | POOL_CLIENT_OUT_OF_MEMORY |
| **Message** | Unable to allocate memory |
| **Description** | System out of memory |
| **Log Module** | pool |
| **Log Group** | internal |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | POOL_CLIENT_DICT_ADD_FAILED |
| **Message** | Failed to add pool client to client dict |
| **Description** | Dictionary addition failed |
| **Log Module** | pool |
| **Log Group** | internal |
| **Severity** | ERROR |

| | |
|---|---|
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | POOL_CLIENT_RANGE_REQ_FAILED |
| **Message** | Failed to send range request to poold error: %s |
| **Description** | Failed to send range request to poold |
| **Log Module** | pool |
| **Log Group** | internal |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | POOL_TABLE_CREATE_FAILED |
| **Message** | Failed to create table, error: %s |
| **Description** | Failed to send range request to poold |
| **Log Module** | pool |
| **Log Group** | internal |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | POOL_CLIENT_DICT_NOT_INITED |
| **Message** | Pool client dictionary is not created yet |
| **Description** | The pool client not found |
| **Log Module** | pool |
| **Log Group** | internal |
| **Severity** | ERROR |
| **Recommended Actions** | Contact Rtbrick customer support |

**Resource Monitoring Log Messages**

**Log Level Summary**

Info(24), Debug(12)

| | |
|---|---|
| **Log ID** | RESMON_RESOURCE_OPTICS_MDL_TEMP_HIGH_ALARM |
| **Message** | Module temperature is above threshold high alarm for module %s |
| **Description** | Resmond resource event logging |
| **Log Module** | resmon |
| **Log Group** | resmon-resource-events |
| **Severity** | INFO |
| **Recommended Actions** | Check resource properties |

| | |
|---|---|
| **Log ID** | RESMON_RESOURCE_OPTICS_MDL_TEMP_HIGH_WARN |
| **Message** | Module temperature is above threshold high warning for module %s |
| **Description** | Resmond resource event logging |
| **Log Module** | resmon |
| **Log Group** | resmon-resource-events |
| **Severity** | INFO |
| **Recommended Actions** | Check resource properties |

| | |
|---|---|
| **Log ID** | RESMON_RESOURCE_OPTICS_MDL_TEMP_LOW_ALARM |
| **Message** | Module temperature is below threshold low alarm for module %s |
| **Description** | Resmond resource event logging |
| **Log Module** | resmon |
| **Log Group** | resmon-resource-events |
| **Severity** | INFO |
| **Recommended Actions** | Check resource properties |

| | |
|---|---|
| **Log ID** | RESMON_RESOURCE_OPTICS_MDL_TEMP_LOW_WARN |
| **Message** | Module temperature is below threshold low warning for module %s |
| **Description** | Resmond resource event logging |

| | |
|---|---|
| **Log Module** | resmon |
| **Log Group** | resmon-resource-events |
| **Severity** | INFO |
| **Recommended Actions** | Check resource properties |

| | |
|---|---|
| **Log ID** | RESMON_RESOURCE_OPTICS_MDL_VOLTAGE_HIGH_ALARM |
| **Message** | Module voltage is above threshold high alarm for module %s |
| **Description** | Resmond resource event logging |
| **Log Module** | resmon |
| **Log Group** | resmon-resource-events |
| **Severity** | INFO |
| **Recommended Actions** | Check resource properties |

| | |
|---|---|
| **Log ID** | RESMON_RESOURCE_OPTICS_MDL_VOLTAGE_HIGH_WARN |
| **Message** | Module voltage is above threshold high warning for module %s |
| **Description** | Resmond resource event logging |
| **Log Module** | resmon |
| **Log Group** | resmon-resource-events |
| **Severity** | INFO |
| **Recommended Actions** | Check resource properties |

| | |
|---|---|
| **Log ID** | RESMON_RESOURCE_OPTICS_MDL_VOLTAGE_LOW_ALARM |
| **Message** | Module voltage is below threshold low alarm for module %s |
| **Description** | Resmond resource event logging |
| **Log Module** | resmon |
| **Log Group** | resmon-resource-events |
| **Severity** | INFO |
| **Recommended Actions** | Check resource properties |

| | |
|---|---|
| **Log ID** | RESMON_RESOURCE_OPTICS_MDL_VOLTAGE_LOW_WARN |
| **Message** | Module voltage is below threshold high alarm for module %s |
| **Description** | Resmond resource event logging |
| **Log Module** | resmon |
| **Log Group** | resmon-resource-events |
| **Severity** | INFO |
| **Recommended Actions** | Check resource properties |

| | |
|---|---|
| **Log ID** | RESMON_RESOURCE_OPTICS_MDL_LANE_POWER_HIGH_ALARM |
| **Message** | Module %s power is above threshold high alarm for module %s on lane %s |
| **Description** | Resmond resource event logging |
| **Log Module** | resmon |
| **Log Group** | resmon-resource-events |
| **Severity** | INFO |
| **Recommended Actions** | Check resource properties |

| | |
|---|---|
| **Log ID** | RESMON_RESOURCE_OPTICS_MDL_LANE_POWER_HIGH_WARN |
| **Message** | Module %s power is above threshold high warning for module %s on lane %s |
| **Description** | Resmond resource event logging |
| **Log Module** | resmon |
| **Log Group** | resmon-resource-events |
| **Severity** | INFO |
| **Recommended Actions** | Check resource properties |

| | |
|---|---|
| **Log ID** | RESMON_RESOURCE_OPTICS_MDL_LANE_POWER_LOW_ALARM |
| **Message** | Module %s power is below threshold low alarm for module %s on lane %s |
| **Description** | Resmond resource event logging |

| | |
|---|---|
| **Log Module** | resmon |
| **Log Group** | resmon-resource-events |
| **Severity** | INFO |
| **Recommended Actions** | Check resource properties |

| | |
|---|---|
| **Log ID** | RESMON_RESOURCE_OPTICS_MDL_LANE_POWER_LOW_WARN |
| **Message** | Module %s power is below threshold high alarm for module %s on lane %s |
| **Description** | Resmond resource event logging |
| **Log Module** | resmon |
| **Log Group** | resmon-resource-events |
| **Severity** | INFO |
| **Recommended Actions** | Check resource properties |

| | |
|---|---|
| **Log ID** | RESMON_RESOURCE_OPTICS_MDL_LANE_BIAS_HIGH_ALARM |
| **Message** | Module %s bias is above threshold high alarm for module %s on lane %s |
| **Description** | Resmond resource event logging |
| **Log Module** | resmon |
| **Log Group** | resmon-resource-events |
| **Severity** | INFO |
| **Recommended Actions** | Check resource properties |

| | |
|---|---|
| **Log ID** | RESMON_RESOURCE_OPTICS_MDL_LANE_BIAS_HIGH_WARN |
| **Message** | Module %s bias is above threshold high warning for module %s on lane %s |
| **Description** | Resmond resource event logging |
| **Log Module** | resmon |
| **Log Group** | resmon-resource-events |
| **Severity** | INFO |

| | |
|---|---|
| **Recommended Actions** | Check resource properties |

| | |
|---|---|
| **Log ID** | RESMON_RESOURCE_OPTICS_MDL_LANE_BIAS_LOW_ALARM |
| **Message** | Module %s bias is below threshold low alarm for module %s on lane %s |
| **Description** | Resmond resource event logging |
| **Log Module** | resmon |
| **Log Group** | resmon-resource-events |
| **Severity** | INFO |
| **Recommended Actions** | Check resource properties |

| | |
|---|---|
| **Log ID** | RESMON_RESOURCE_OPTICS_MDL_LANE_BIAS_LOW_WARN |
| **Message** | Module %s bias is below threshold high alarm for module %s on lane %s |
| **Description** | Resmond resource event logging |
| **Log Module** | resmon |
| **Log Group** | resmon-resource-events |
| **Severity** | INFO |
| **Recommended Actions** | Check resource properties |

| | |
|---|---|
| **Log ID** | RESMON_RESOURCE_OPTICS_MDL_LANE_LOS |
| **Message** | Module %s loss of signal alarm for module %s on lane %s |
| **Description** | Resmond resource event logging |
| **Log Module** | resmon |
| **Log Group** | resmon-resource-events |
| **Severity** | INFO |
| **Recommended Actions** | Check resource properties |

| | |
|---|---|
| **Log ID** | RESMON_RESOURCE_OPTICS_MDL_LANE_LOL |
| **Message** | Module %s loss of light alarm for module %s on lane %s |

| | |
|---|---|
| **Description** | Resmond resource event logging |
| **Log Module** | resmon |
| **Log Group** | resmon-resource-events |
| **Severity** | INFO |
| **Recommended Actions** | Check resource properties |

| | |
|---|---|
| **Log ID** | RESMON_NTP_PEER_DOWN |
| **Message** | NTP Peer down ID %s with IP %s resolving to %s |
| **Description** | Resmond ntp peer event logging |
| **Log Module** | resmon |
| **Log Group** | resmon-ntp-peer-events |
| **Severity** | INFO |
| **Recommended Actions** | Check resource properties |

| | |
|---|---|
| **Log ID** | RESMON_NTP_PEER_UP |
| **Message** | NTP Peer up ID %s with IP %s resolving to %s |
| **Description** | Resmond ntp peer event logging |
| **Log Module** | resmon |
| **Log Group** | resmon-ntp-peer-events |
| **Severity** | INFO |
| **Recommended Actions** | Check resource properties |

| | |
|---|---|
| **Log ID** | RESMON_NTP_PEER_BEST |
| **Message** | NTP Peer selected as best peer for sync ID %s with IP %s resolving to %s |
| **Description** | Resmond ntp peer event logging |
| **Log Module** | resmon |
| **Log Group** | resmon-ntp-peer-events |
| **Severity** | INFO |
| **Recommended Actions** | Check resource properties |

| | |
|---|---|
| **Log ID** | RESMON_NTP_PEER_NOT_BEST |
| **Message** | NTP Peer deselected as best peer for sync ID %s with IP %s resolving to %s |
| **Description** | Resmond ntp peer event logging |
| **Log Module** | resmon |
| **Log Group** | resmon-ntp-peer-events |
| **Severity** | INFO |
| **Recommended Actions** | Check resource properties |

| | |
|---|---|
| **Log ID** | RESMON_NTP_PEER_UNREACHABLE |
| **Message** | NTP Peer unreachable ID %s with IP %s resolving to %s |
| **Description** | Resmond ntp peer event logging |
| **Log Module** | resmon |
| **Log Group** | resmon-ntp-peer-events |
| **Severity** | INFO |
| **Recommended Actions** | Check resource properties |

| | |
|---|---|
| **Log ID** | RESMON_NTP_PEER_REACHABLE |
| **Message** | NTP Peer reachable ID %s with IP %s resolving to %s |
| **Description** | Resmond ntp peer event logging |
| **Log Module** | resmon |
| **Log Group** | resmon-ntp-peer-events |
| **Severity** | INFO |
| **Recommended Actions** | Check resource properties |

| | |
|---|---|
| **Log ID** | RESMON_GENERIC |
| **Message** | RESMOND generic event - %s |
| **Description** | Resmond generic event logging |
| **Log Module** | resmon |

| | |
|---|---|
| **Log Group** | resmon-internal-events |
| **Severity** | DEBUG |
| **Recommended Actions** | Check resmond resource and intenal tables |

| | |
|---|---|
| **Log ID** | RESMON_GENERIC_RC |
| **Message** | RESMOND generic event - %s status[%s - %s] |
| **Description** | Resmond generic event logging |
| **Log Module** | resmon |
| **Log Group** | resmon-internal-events |
| **Severity** | DEBUG |
| **Recommended Actions** | Check resmond resource and intenal tables |

| | |
|---|---|
| **Log ID** | RESMON_BDS_OBJ_ALLOC_ERR |
| **Message** | Unable to allocate object template for %s |
| **Description** | Resmond bds event logging |
| **Log Module** | resmon |
| **Log Group** | resmon-bds-events |
| **Severity** | DEBUG |
| **Recommended Actions** | Check resmond resource and intenal tables |

| | |
|---|---|
| **Log ID** | RESMON_BDS_ATTR_NOT_FOUND |
| **Message** | Expecting attribute "%s", but not found |
| **Description** | Resmond bds event logging |
| **Log Module** | resmon |
| **Log Group** | resmon-bds-events |
| **Severity** | DEBUG |
| **Recommended Actions** | Check resmond resource and intenal tables |

| | |
|---|---|
| **Log ID** | RESMON_BDS_ATTR_OBJ_NOT_FOUND |
| **Message** | Unable to get attribute "%s" in "%s" |
| **Description** | Resmond bds event logging |
| **Log Module** | resmon |
| **Log Group** | resmon-bds-events |
| **Severity** | DEBUG |
| **Recommended Actions** | Check resmond resource and intenal tables |

| | |
|---|---|
| **Log ID** | RESMON_BDS_TABLE_NOT_FOUND |
| **Message** | Table "%s" not present |
| **Description** | Resmond bds event logging |
| **Log Module** | resmon |
| **Log Group** | resmon-bds-events |
| **Severity** | DEBUG |
| **Recommended Actions** | Check resmond resource and intenal tables |

| | |
|---|---|
| **Log ID** | RESMON_BDS_TBL_CREATE_ERR |
| **Message** | Failed to create table %s of type %s |
| **Description** | Resmond bds event logging |
| **Log Module** | resmon |
| **Log Group** | resmon-bds-events |
| **Severity** | DEBUG |
| **Recommended Actions** | Check resmond resource and intenal tables |

| | |
|---|---|
| **Log ID** | RESMON_BDS_TBL_SUBSCRIBE_ERR |
| **Message** | Failed to subscribe table %s |
| **Description** | Resmond bds event logging |
| **Log Module** | resmon |
| **Log Group** | resmon-bds-events |
| **Severity** | DEBUG |

| | |
|---|---|
| **Recommended Actions** | Check resmond resource and intenal tables |

| | |
|---|---|
| **Log ID** | RESMON_INTERNAL_FILE_OPEN_ERR |
| **Message** | File %s open failed[%s - %s] |
| **Description** | Resmond internal event logging |
| **Log Module** | resmon |
| **Log Group** | resmon-internal-events |
| **Severity** | DEBUG |
| **Recommended Actions** | Check resmond resource and intenal tables |

| | |
|---|---|
| **Log ID** | RESMON_INTERNAL_FILE_CLOSE_ERR |
| **Message** | File %s open failed[%s - %s] |
| **Description** | Resmond internal event logging |
| **Log Module** | resmon |
| **Log Group** | resmon-internal-events |
| **Severity** | DEBUG |
| **Recommended Actions** | Check resmond resource and intenal tables |

| | |
|---|---|
| **Log ID** | RESMON_INTERNAL_FILE_NOT_FOUND |
| **Message** | File %s not found |
| **Description** | Resmond internal event logging |
| **Log Module** | resmon |
| **Log Group** | resmon-internal-events |
| **Severity** | DEBUG |
| **Recommended Actions** | Check resmond resource and intenal tables |

| | |
|---|---|
| **Log ID** | RESMON_INTERNAL_CMD_EXEC_FAIL |
| **Message** | Failed to execute system cmd[%s], err code[%s] |

| | |
|---|---|
| **Description** | Resmond internal event logging |
| **Log Module** | resmon |
| **Log Group** | resmon-internal-events |
| **Severity** | DEBUG |
| **Recommended Actions** | Check resmond resource and intenal tables |

### validation Log Messages

#### Log Level Summary

Debug(33)

| | |
|---|---|
| **Log ID** | VALIDATION_TABLE_CREATED |
| **Message** | BDS table %s created |
| **Description** | BDS table created |
| **Log Module** | validation |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | VALIDATION_TABLE_CREATE_ERROR |
| **Message** | Failed to create BDS table %s |
| **Description** | Failed to create BDS table |
| **Log Module** | validation |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | VALIDATION_TABLE_DELETED |
| **Message** | BDS table %s deleted |
| **Description** | BDS table deleted |
| **Log Module** | validation |

| | |
|---|---|
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | VALIDATION_TABLE_ATTR_NOT_FOUND |
| **Message** | Failed to get table attribute for table %s |
| **Description** | BDS table attribute not found |
| **Log Module** | validation |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | VALIDATION_TABLE_NOT_FOUND |
| **Message** | Failed to get table %s |
| **Description** | BDS table not found |
| **Log Module** | validation |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | VALIDATION_TABLE_SUBSCRIBED |
| **Message** | BDS table %s subscribed |
| **Description** | BDS table subscribed |
| **Log Module** | validation |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | VALIDATION_TABLE_UNSUBSCRIBED |
| **Message** | BDS table %s unsubscribed |
| **Description** | BDS table unsubscribed |
| **Log Module** | validation |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | VALIDATION_TRAP_ADDED |
| **Message** | Trap %s added |
| **Description** | Validation trap added |
| **Log Module** | validation |
| **Log Group** | config |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | VALIDATION_TRAP_DELETED |
| **Message** | Trap %s deleted |
| **Description** | Validation trap deleted |
| **Log Module** | validation |
| **Log Group** | config |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | VALIDATION_OBJ_GET_FAILED |
| **Message** | Failed to get object for table %s |
| **Description** | BDS object get failed |
| **Log Module** | validation |
| **Log Group** | general |
| **Severity** | debug |

| | |
|---|---|
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | VALIDATION_OBJ_ADD_FAILED |
| **Message** | Failed to add object for table %s |
| **Description** | BDS object add failed |
| **Log Module** | validation |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | VALIDATION_OBJ_DEL_FAILED |
| **Message** | Failed to delete object for table %s |
| **Description** | BDS object delete failed |
| **Log Module** | validation |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | VALIDATION_CACHE_ADDED |
| **Message** | Cache object added in instance %s with cache-ip %s src-ip %s |
| **Description** | Cache object added |
| **Log Module** | validation |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | VALIDATION_CACHE_DELETED |
| **Message** | Cache object deleted from instance %s with cache-ip %s src-ip %s |

| | |
|---|---|
| **Description** | Cache object deleted |
| **Log Module** | validation |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | VALIDATION_CACHE_ADD_FAILED |
| **Message** | Failed to add cache object in instance %s with cache-ip %s src-ip %s |
| **Description** | Cache object add failed |
| **Log Module** | validation |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | VALIDATION_CACHE_DEL_FAILED |
| **Message** | Failed to delete cache object from instance %s with cache-ip %s src-ip %s |
| **Description** | Cache object delete failed |
| **Log Module** | validation |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | VALIDATION_CACHE_CHANGED |
| **Message** | Cache object changed in instance %s with cache-ip %s src-ip %s |
| **Description** | Cache object changed |
| **Log Module** | validation |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | VALIDATION_CACHE_GET_FAILED |
| **Message** | Failed to get cache in instance %s with cache-ip %s src-ip %s |
| **Description** | Cache object get failed |
| **Log Module** | validation |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | VALIDATION_CACHE_STOPPED |
| **Message** | Cache stopped in instance %s with cache-ip %s src-ip %s |
| **Description** | Cache stopped |
| **Log Module** | validation |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | VALIDATION_CACHE_FLUSHED |
| **Message** | Cache stale object flushed in instance %s with cache-ip %s src-ip %s |
| **Description** | Cache stale object flushed |
| **Log Module** | validation |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | VALIDATION_CACHE_FLUSH_COUNT |
| **Message** | %s stale objects flushed in table %s |
| **Description** | Cache stale object flushed |
| **Log Module** | validation |

| | |
|---|---|
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | VALIDATION_CACHE_DATABASE_FLUSH |
| **Message** | full table %s flushed |
| **Description** | Cache database flushed |
| **Log Module** | validation |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | VALIDATION_CACHE_START |
| **Message** | Tring to start the cache in instance %s with cache-ip %s src-ip %s |
| **Description** | Cache start |
| **Log Module** | validation |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | VALIDATION_CACHE_CONNECT |
| **Message** | Tring to connect to cache in instance %s with cache-ip %s src-ip %s port %s |
| **Description** | Cache connect |
| **Log Module** | validation |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | VALIDATION_CACHE_CONNECT_FAILED |
| **Message** | Connect failed to cache in instance %s with cache-ip %s src-ip %s port %s |
| **Description** | Cache connect failed |
| **Log Module** | validation |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | VALIDATION_PKT_ERROR |
| **Message** | Error while processing pdu %s from cache in instance %s with cache-ip %s src-ip %s |
| **Description** | Cache packet error |
| **Log Module** | validation |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | VALIDATION_PKT_ERROR_SEND |
| **Message** | Sending error %s %s to cache in instance %s with cache-ip %s src-ip %s |
| **Description** | Cache packet error |
| **Log Module** | validation |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | VALIDATION_PKT_ERROR_CACHE_NOT_FOUND |
| **Message** | Failed to get cache during pkt processing in instance %s with cache-ip %s src-ip %s |
| **Description** | Cache object not found |
| **Log Module** | validation |

| | |
|---|---|
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | VALIDATION_FSM_TIMER_STARTED |
| **Message** | Timer %s started for cache in instance %s with cache-ip %s src-ip %s |
| **Description** | Cache fsm timer started |
| **Log Module** | validation |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | VALIDATION_FSM_TIMER_STOPPED |
| **Message** | Timer %s stopped for cache in instance %s with cache-ip %s src-ip %s |
| **Description** | Cache fsm timer stopped |
| **Log Module** | validation |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | VALIDATION_FSM_TIMER_EXPIRED |
| **Message** | Timer %s expired for cache in instance %s with cache-ip %s src-ip %s |
| **Description** | Cache fsm timer expired |
| **Log Module** | validation |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | VALIDATION_FSM_EVENT |
| **Message** | cache [%s cache-ip %s src-ip %s] moved from %s to %s with event %s |
| **Description** | Cache fsm timer expired |
| **Log Module** | validation |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

| | |
|---|---|
| **Log ID** | VALIDATION_LWIP_EVENT |
| **Message** | Event %s for cache [%s cache-ip %s src-ip %s] |
| **Description** | Cache lwip event |
| **Log Module** | validation |
| **Log Group** | general |
| **Severity** | debug |
| **Recommended Actions** | Contact Rtbrick customer support |

# 11.4. LED Control

## 11.4.1. LED Overview

You can use the switch LEDs to monitor the activity and performance of a network switch or router. Using the LED control functionality, you can control the LEDs that are available in the hardware platforms supported by RBFS.

There are two types of LEDs available in the switch:

- Network Port LEDs
- System LEDs

**Supported Platforms**

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

## 11.4.2. LED Definitions

### Port LED Definition for UfiSpace S9510-28DC

The network port LEDs are used to represent the status of the link associated with the LED.

| Port type | Condition | Status |
|---|---|---|
| SFP+ | Up | Green |
| | Down | Off |
| SFP28 | Up | Green |
| | Down | Off |
| QSFP28 | Up | Green |
| | Down | Off |

### System LED Definition for UfiSpace S9510-28DC

The system LEDs are used to indicate the status of power and system.

| LED Indicators | Behavior | Status |
|---|---|---|
| Power | Off | No power or in shut down mode. |
| | Solid Green | System power is good. |
| | Blinking Green | System power is good but BMC power is failed. |
| | Solid Yellow | System power is good but CPU power is failed. |
| | Blinking Yellow | System power failure. |
| STAT (System Status) | Off | System (X86 and BMC) is not booted. |
| | Solid Green | System boot is complete. |
| | Blinking Green | System boot is in progress. |
| | Solid Yellow | System boot is complete. (DIAG OS) |
| | Blinking Yellow | Reserved |
| PSU LED Functions and State | Off | No DC power to any of the PSUs. |
| | Flashing Red | No DC power to this particular PSU. |
| | Flashing Green | DC is present, only standby output is on. Poor contact. |
| | Green | PSU output is on and normal. |
| | Red | PSU failure. |
| | Flashing between Green and Red | Warning. Working condition is not satisfied. Check the voltage, electric current, and temperature. |

| LED Indicators | Behavior | Status |
|---|---|---|
| FAN LED Functions and State | Off | Main board 3.3V power is failed or fan is not present. |
| | Solid Green | Fan is present but interrupt assert |
| | Blinking Green | NA |
| | Solid Yellow | NA |
| | Blinking Yellow | Fan is present but interrupt assert. |

**Port LED Definition for Edgecore CSR440**

The network port LEDs are used to represent the status of the link associated with the LED.

| Port type | Condition | Status |
|---|---|---|
| 10G | Up | Green |
| | Down | LED is Off |
| 25G | Up | Blue |
| | Down | LED is Off |
| 100G | Up | Two LEDs per port. Green (on QSFP28) and Violet (on QSFP-DD). |
| | Down | LED is off |

**System LED Definition for Edgecore CSR440**

The system LEDs are used to indicate the status of power and system.

| Port type | Condition | Status |
|---|---|---|
| PSU1 (power supply unit) | Solid Green | PSU1 is operating normally. |
| | Solid Amber | Input power is present, but the PSU1 is not ON or is in a faulty state. |
| | Off | The input power is not present or PSU1 is not present. |
| PSU2 (power supply unit) | Solid Green | PSU2 is operating normally. |
| | Solid Amber | Input power is present, but the PSU2 is not ON or is in a faulty state. |
| | Off | Input power is not present or PSU2 is not present. |

| Port type | Condition | Status |
|---|---|---|
| DIAG (Diagnostic) | Solid Green | System self diagnostic test is successfully completed. |
| | Blinking Green | System self-diagnostic test is in progress. |
| | Solid Orange | System self-diagnostic test has detected a fault. |
| FAN | Solid Green | All fan modules are present and working normally. |
| | Solid Orange | Fan module is present but a fault has been detected. |
| | Off | Fan module is not present. |
| LOC | Blinking Blue | Flashing by remote management command. It helps technicians in finding the right device for service in the rack. It provides an indication of the location of the device. |
| | Off | No indication. |
| ALARM | Solid Green | No alarms detected. |
| | Solid Red | Local alarm is detected. |

**Port LED Definition for UfiSpace S9600-72XC**

The network port LEDs are used to show the status of the link associated with the LED. In an UfiSpace S9600-72XC switch, there are 64*25G SFP28 ports and 8*100G QSFP28 on the front panel. SFP28 and QSFP28 operate in full duplex mode when the speed is 25Gbps and 100Gbps respectively.

| LED | Condition | Status |
|---|---|---|
| SFP28 Port LED | On (Green) | Link is up |
| | Off | No link |
| QSFP28 Port LED | On (Green) | Link is up |
| | Off | No link |

**System LED Definition for UfiSpace S9600-72XC**

The system LEDs are used to indicate the status of power and system.

| LED | Condition | Status |
|---|---|---|
| SYS | Off | No power |
| | Solid Green | Host CPU/BMC boot is complete |
| | Solid Amber | Power is up, but host CPU/BMC boot is failed. |
| FAN | Off | Fans are not initialized. |
| | Solid Green | All fans are working normally. |
| | Blinking Amber | Fan is not working. One or more fans need service. |
| PSU1 | Off | No power |
| | Solid Green | PSU1 is working normally. |
| | Blinking Amber | PSU1 is failed. Service required. |
| PSU2 | Off | No power |
| | Solid Green | PSU2 is working normally. |
| | Blinking Amber | PSU2 is failed. Service is needed. |

**Port LED Definition for EdgeCore AGR420**

The network port LEDs are used to represent the status of the link associated with the LED.

| Port type | Condition | Status |
|---|---|---|
| SFP+ | Up | Blue LED is on (2 LEDs per port) |
| | Down | Off |
| SFP28 | Up | Blue LED is on (2 LEDs per port) |
| | Down | Off |
| QSFP28 | Up | Blue LED is on (2 LEDs per port) |
| | Down | Off |

**System LED Definition for Edgecore AGR420**

The system LEDs are used to indicate the status of power and system.

| LED | Condition | Status |
|---|---|---|
| DIAG (Diagnostic) | Green | System self diagnostic test is successfully completed. |
| | Amber | System self-diagnostic test has detected a fault. (Faulty Fan, thermal, or any interface.) |

| LED | Condition | Status |
|---|---|---|
| LOC | Flashing Amber | Flashing by remote management command. Assists technician in finding the right device for service in the rack. |
| | Off | Not the switch that technician needs to find. |
| FAN | Green | System fan is working normally. |
| | Amber | Fan tray is present but system fan is faulty. |
| | Off | System is off. |
| PSU0 (power supply status) | Green | The power supply, #0, is working normally. |
| | Amber | Power is present, but not ON. Power is faulty. |
| | Off | Power supply is not present. |
| PSU1 (power supply status) | Green | The power supply, #1, is working normally. |
| | Amber | Power is present, but not ON. Power is faulty. |
| | Off | Power supply is not present. |

**Port LED Definition for UfiSpace S9600-32X**

The network port LEDs are used to show the status of the link associated with the LED. In a UfiSpace S9600-32X switch, there are 4*10G SFP+ ports and 32*100G QSFP28 ports on the front panel. The SFP+ and QSFP28 ports operate in full duplex mode when the speed is 10Gbps and 100Gbps respectively.

| LED | Condition | Status |
|---|---|---|
| SFP+ Port LED | On (Green) | Link is up |
| | Off | No link |
| QSFP28 Port LED | On (Green) | Link is up |
| | Off | No link |

**System LED Definition for UfiSpace S9600-32X**

The system LEDs are used to indicate the status of power and system.

| LED | Condition | Status |
|---|---|---|
| SYS | Off | System (x86 and BMC) is not booted. |
| | Solid Green | System boot is complete. |
| | Blinking Green | System boot is in progress. |
| | Fast Blinking Green | Reserved. |
| | Solid Amber | Power is up but system boot is failed. |
| | Blinking Amber | Reserved |
| | Fast Blinking Amber | Reserved |
| FAN | Off | Main board 3.3V_FAN power failure or no power. |
| | Solid Green | All fans are working normally and interrupt de-assert. |
| | Blinking Green | Fans PWMs do not match or each one fan no present |
| | Fast Blinking Green | Reserved. |
| | Solid Amber | Each one of fans is failure (PWM=0) |
| | Blinking Amber | Fan is present but interrupt assert. |
| | Fast Blinking Amber | Reserved |
| PSU1 | Off | No DC power or standby is failed. |
| | Solid Green | PSU DC output is ON and standby is normal. |
| | Blinking Green | No PSU is present. |
| | Fast Blinking Green | Reserved. |
| | Solid Amber | PSU DC output is failed and standby is normal. |
| | Blinking Amber | Warning. Working condition is not satisfied. (Check the voltage, electric current, and temperature). |
| | Fast Blinking Amber | Reserved |

| LED | Condition | Status |
|---|---|---|
| PSU2 | Off | No DC power or standby is failed. |
| | Solid Green | PSU DC output is ON and standby is normal. |
| | Blinking Green | No PSU is present. |
| | Fast Blinking Green | Reserved. |
| | Solid Amber | PSU DC output is failed and standby is normal. |
| | Blinking Amber | Warning. Working condition is not satisfied. (Check the voltage, electric current, and temperature). |
| | Fast Blinking Amber | Reserved |
| ID | Off | Turned off location of function |
| | Solid Blue | Ready for location of function |
| | Blinking Blue | Location of switch in telcom center |
| | Fast Blinking Blue | Reserved. |

**Port LED Definition for Edgecore AGR400**

The network port LEDs are used to represent the status of the link associated with the LED.

| Port type | Condition | Status |
|---|---|---|
| SFP28 | Up | Blue LED is on (Single LED per port) |
| | Down | LED is Off |
| QSFP28 | Up | Blue LED is on (2 LEDs per port) |
| | Down | LED is Off |
| QSFP-DD | Up | Not tested |
| | Down | Not tested |

**System LED Definition for Edgecore AGR400**

The system LEDs are used to indicate the status of power and system.

| LED | Condition | Status |
|---|---|---|
| DIAG (Diagnostic) | Green | System self diagnostic test is successfully completed. |
| | Amber | System self-diagnostic test has detected a fault. (Faulty Fan, thermal, or any interface.) |
| LOC | Flashing Amber | Flashing by remote management command. Assists the technician in finding the right device for service in the rack. |
| | Off | Not the switch that the technician needs to find. |
| FAN | Green | System fan is working normally. |
| | Amber | Fan tray is present but system fan is faulty. |
| | Off | System is off. |
| PSU0 (power supply status) | Green | The power supply, #0, is working normally. |
| | Amber | Power is present, but not ON. Power is faulty. |
| | Off | Power supply is not present. |
| PSU1 (power supply status) | Green | The power supply, #1, is working normally. |
| | Amber | Power is present, but not ON. Power is faulty. |
| | Off | Power supply is not present. |

**Port LED Definition for UfiSpace S9610-46DX**

The network port LEDs indicate the operational status of the associated interface. The UfiSpace S9610-46DX provides 40 QSFP28 (40/100G) ports, 6 QSFP-DD (100/200/400G) ports, and 4 SFP28/SFP-DD (1/10/25G) ports. All interfaces operate in full-duplex mode at their negotiated link speed.

| Port Type & Speed | LED State | Status |
|---|---|---|
| SFP-DD 25G | Green Solid | Link up |
| | Blinking Green | Packet transmitting or receiving |
| | Off | No link or port disabled |
| SFP-DD 10G/1G | Amber Solid | Link up |
| | Blinking Amber | Packet transmitting or receiving |
| | Off | No link or port disabled |

| Port Type & Speed | LED State | Status |
|---|---|---|
| QSFP28 Non-breakout mode: 100G | Green Solid | Link up |
| | Blinking Green | Packet transmitting or receiving |
| | Off | No link or port disabled |
| QSFP28 Non-breakout mode: 40G | Amber Solid | Link up |
| | Blinking Amber | Packet transmitting or receiving |
| | Off | No link or port disabled |
| QSFP28 Breakout mode: 25G/port | Green Solid | All break-out ports link up |
| | Blinking Green | All break-out ports have traffic activity |
| | Off | All break-out ports link down |
| | Amber Solid | Any one break-out port link down |
| | Blinking Amber | Any break-out port has traffic activity |
| | Off | All break-out ports link down |
| QSFP-DD Non-breakout mode: 400G | Green Solid | Link up |
| | Blinking Green | Packet transmitting or receiving |
| | Off | No link or port disabled |
| QSFP-DD Non-breakout mode: 100G | Amber Solid | Link up |
| | Blinking Amber | Packet transmitting or receiving |
| | Off | No link or port disabled |
| QSFP-DD Breakout mode: 50G/port | Solid Green | All break-out ports link up |
| | Blinking Green | All break-out ports have traffic activity |
| | Off | All break-out ports link down |
| | Solid Amber | Any one break-out port link down |
| | Blinking Amber | Any break-out port has traffic activity |
| | Off | All break-out ports link down |

**System LED Definition for UfiSpace S9610-46DX**

The system LEDs are used to indicate the status of power and system.

| LED | Condition | Status |
|---|---|---|
| SYS | OFF | System (X86 & BMC) no boot |
| | Solid Green | System boot complete (NOS) |
| | Blinking Green | Reserved |
| | Solid Yellow | Power is up but Host CPU/BMC boot failed |
| | Blinking Yellow | Reserved |

| LED | Condition | Status |
|---|---|---|
| FAN | OFF | Fans are not initialized |
| | Solid Green | All FAN modules work well |
| | Blinking Green | Reserved |
| | Solid Yellow | Reserved |
| | Blinking Yellow | One or more FAN module(s) fail or no FAN module present |
| PS0 | OFF | No power |
| | Solid Green | PSU0 is work normal |
| | Blinking Green | Reserved |
| | Solid Yellow | Reserved |
| | Blinking Yellow | PSU0 fail (PSU0 need service) |
| PS1 | OFF | No power |
| | Solid Green | PSU1 is work normal |
| | Blinking Green | Reserved |
| | Solid Yellow | Reserved |
| | Blinking Yellow | PSU1 fail (PSU1 need service) |
| SYNC | OFF | System timing synchronization is disabled or in free-run mode. |
| | Solid Green | System timing core is synchronized to PTP packet (phase) or one of external timing sources, frequency is also locked. |
| | Blinking Green | System is synchronized to PTP packet (frequency) or one of external timing (frequency) sources. |
| | Solid Yellow | System timing core is in acquiring state or holdover mode. |
| | Blinking Yellow | System timing synchronization fail, or acquiring/hold over time out. |
| ID | OFF | No power |
| | Solid Blue | Reserved |
| | Blinking Blue | Enable beacon feature |

**Port LED Definition for UfiSpace S9500-22XST**

The network port LEDs are used to show the status of the link associated with the LED. In a UfiSpace S9500-22XST switch, there are 12*10G SFP+ ports, 8*25G SFP28 ports, and 2*100G QSFP28 ports on the front panel. The SFP, SFP28, and QSFP28 ports operate in a full duplex mode when the speed is 10Gbps, 25Gbps, or 100Gbps.

| LED | Condition | Status |
|---|---|---|
| SFP+ Port LED | On (Green) | Link is up |
| | Off | No link |
| SFP28 Port LED | On (Green) | Link is up |
| | Off | No link |
| QSFP28 Port LED | On (Green) | Link is up |
| | Off | No link |

**System LED Definition for UfiSpace S9500-22XST**

The system LEDs are used to indicate the status of power and system.

| LED Indicators | Condition | Status |
|---|---|---|
| Power | Off | No power or in shut down mode. |
| | Solid Green | System power is good and the BMC heating is complete. |
| | Blinking Green | System power is good and the BMC heating is in progress. |
| | Solid Yellow | System power is good but BMC heating is failed. |
| | Blinking Yellow | System power failure. |
| STAT (System Status) | Off | System (X86 and BMC) is no booted. |
| | Solid Green | System boot is complete. |
| | Blinking Green | System boot is in progress. |
| | Solid Yellow | Reserved |
| | Blinking Yellow | Reserved |
| PSU LED Functions and State | Off | No DC power to any of the PSUs. |
| | Flashing Red | No DC power to this particular PSU. |
| | Flashing Green | DC is present, only standby output is on. Poor contact. |
| | Green | PSU DC output is on and is normal. |
| | Red | PSU failure. |
| | Flashing between Green and Red | Warning. Working condition is not satisfied. Check the voltage, electric current, and temperature. |

| LED Indicators | Condition | Status |
|---|---|---|
| FAN LED Functions and State | Off | Main board 3.3V power is failed or fan is not present. |
| | Solid Green | Fan is present and interrupt de-assert. |
| | Blinking Green | NA |
| | Solid Yellow | NA |
| | Blinking Yellow | Fan is present but interrupt assert. |

# 11.5. IPMI

## 11.5.1. IPMI Overview

The IPMI (Intelligent Platform Management Interface) is an open-standard hardware management interface that allows you communicate with BMC (Baseboard Management Controller) of the platform hardware.

BMC, a dedicated micro-controller, manages the interface between system-management software and platform hardware. BMC performs operations such as remote power on or power off, or makes the console accessible. BMC also monitors the hardware resources such as sensors and can send alert messages over the LAN indicating a potential failure of the system.

IPMI is a way to manage the platform hardware by interacting with the hardware directly rather than with the operating system. The advantages of using IPMI are that it allows an out-of-band platform hardware management and the operating system is not burdened with the operational overheads by sending system status data.

### Supported Platforms

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

## 11.5.2. IPMI Configuration

### Configuration Syntax and Commands

The following sections describe the IPMI configuration syntax and commands.

#### IPMI Interface Configuration

This configuration allows you to communicate with the BMC on the platform hardware.

**Syntax:**

```
set system platform-management ipmi interface <interface-id> <attribute> <value>
```

| Attribute | Description |
|---|---|
| `interface <interface-id>` | IPMI channel ID. For UfiSpace switches, the LAN interface ID is 1. |
| `type <lan>` | type of network |
| `mode <dhcp\|static>` | Manually set the static IPv4 address or configure DHCP to receive dynamic IPv4 address. |
| `address-family ipv4 prefix4 <prefix4>` | IPv4 prefix |
| `address-family ipv4 gateway-address <gateway-address>` | IPv4 address for gateway |

Example: IPMI Interface Configuration

```
{
  "ietf-restconf:data": {
    "rtbrick-config:system": {
      "platform-management": {
        "ipmi": {
          "interface": [
            {
              "id": 1,
              "type": "lan",
              "mode": "static",
              "address-family": {
                "ipv4": {
                  "prefix4": "198.51.100.110/24",
                  "gateway-address": "198.51.100.111"
                }
              }
            }
          ]
        }
      }
    }
  }
}
```

#### IPMI User Configuration

This configuration allows you to change the password of one of the 10 users, including the admin user.

**Syntax:**

**set system platform-management ipmi user** <user-id> <options>

| Attribute | Description ` |
|---|---|
| user <user-id>` | User ID. The `ipmi user id` of the default 'admin' user on UfiSpace switches is 2. |
| `password-plain-text <password-plain-text>` | Plain text password |
| `password-encrypted-text <password-encrypted-text>` | Encrypted password |

Example: IPMI User Configuration

```
{
  "ietf-restconf:data": {
    "rtbrick-config:system": {
      "platform-management": {
        "ipmi": {
          "user": [
            {
              "id": 2,
              "password-encrypted-text": "$21e4946e31b406de98b3077aef03ed5a7"
            }
          ]
        }
      }
    }
  }
}
```

## 11.5.3. IPMI Operational Commands

### IPMI Validation Commands

The IPMI validation commands provide you the IPMI configuration, channel, and user information. Configure the server that connects to the hardware platform. Use the IPMI tool to verify the remote accessibility by the running the following commands on the Linux shell.

Verification of remote access can be performed using the following command:

`ipmitool -I lanplus -H <ip address> -U admin -P admin lan print 1`

Example for remote access verification

```
supervisor@srv10-tst:~$ ipmitool -I lanplus -H 198.51.100.100 -U admin -P admin lan
print 1
Set in Progress         : Set Complete
Auth Type Support       :
Auth Type Enable        : Callback : MD5
                        : User     : MD5
                        : Operator : MD5
```

```
                        : Admin    : MD5
                        : OEM      : MD5
IP Address Source       : Static Address
IP Address              : 198.51.100.100
Subnet Mask             : 255.255.255.128
MAC Address             : e8:c5:7a:8f:78:0d
SNMP Community String   : AMI
IP Header               : TTL=0x40 Flags=0x40 Precedence=0x00 TOS=0x10
BMC ARP Control         : ARP Responses Enabled, Gratuitous ARP Disabled
Gratituous ARP Intrvl   : 0.0 seconds
Default Gateway IP      : 198.51.100.41
Default Gateway MAC     : 00:00:5e:00:01:01
Backup Gateway IP       : 198.51.100.10
Backup Gateway MAC      : 00:00:00:00:00:00
802.1q VLAN ID          : Disabled
802.1q VLAN Priority    : 0
RMCP+ Cipher Suites     : 0,1,2,3,6,7,8,11,12,15,16,17
Cipher Suite Priv Max   : caaaaaaaaaaaXXX
                        :     X=Cipher Suite Unused
                        :     c=CALLBACK
                        :     u=USER
                        :     o=OPERATOR
                        :     a=ADMIN
                        :     O=OEM
Bad Password Threshold  : 0
Invalid password disable: no
Attempt Count Reset Int.: 0
User Lockout Interval   : 0
```

Use the following command to verify the channel information:

`sudo ipmitool channel info 1`

Example for channel information

```
supervisor@onl>ufi06.q2c.u25.r4.nbg.rtbrick.net:~ $ sudo ipmitool channel info 1
Channel 0x1 info:
  Channel Medium Type   : 802.3 LAN
  Channel Protocol Type : IPMB-1.0
  Session Support       : multi-session
  Active Session Count  : 0
  Protocol Vendor ID    : 7154
  Volatile(active) Settings
    Alerting            : enabled
    Per-message Auth    : disabled
    User Level Auth     : enabled
    Access Mode         : always available
  Non-Volatile Settings
    Alerting            : enabled
    Per-message Auth    : disabled
```

```
    User Level Auth     : enabled
    Access Mode         : always available
```

Use the following command to verify a specific channel information:

`sudo ipmitool lan print 1`

Example for specific channel information

```
supervisor@onl>ufi06.q2c.u25.r4.nbg.rtbrick.net:~ $ sudo ipmitool lan print 1
Set in Progress         : Set Complete
Auth Type Support       :
Auth Type Enable        : Callback : MD5
                        : User     : MD5
                        : Operator : MD5
                        : Admin    : MD5
                        : OEM      : MD5
IP Address Source       : Static Address
IP Address              : 198.51.100.100
Subnet Mask             : 255.255.255.128
MAC Address             : e8:c5:7a:8f:78:0d
SNMP Community String   : AMI
IP Header               : TTL=0x40 Flags=0x40 Precedence=0x00 TOS=0x10
BMC ARP Control         : ARP Responses Enabled, Gratuitous ARP Disabled
Gratituous ARP Intrvl   : 0.0 seconds
Default Gateway IP      : 198.51.100.41
Default Gateway MAC     : 00:00:5e:00:01:01
Backup Gateway IP       : 198.51.100.10
Backup Gateway MAC      : 00:00:00:00:00:00
802.1q VLAN ID          : Disabled
802.1q VLAN Priority    : 0
RMCP+ Cipher Suites     : 0,1,2,3,6,7,8,11,12,15,16,17
Cipher Suite Priv Max   : caaaaaaaaaaaXXX
                        :     X=Cipher Suite Unused
                        :     c=CALLBACK
                        :     u=USER
                        :     o=OPERATOR
                        :     a=ADMIN
                        :     O=OEM
Bad Password Threshold  : 0
Invalid password disable: no
Attempt Count Reset Int.: 0
User Lockout Interval   : 0
```

Use the following command to verify the user for a channel:

`sudo ipmitool user list 1`

Example for user per channel

```
supervisor@onl>ufi06.q2c.u25.r4.nbg.rtbrick.net:~ $ sudo ipmitool user list 1
ID  Name     Callin  Link Auth IPMI Msg   Channel Priv Limit
1                    false   false      true       ADMINISTRATOR
2   admin            false   false      true       ADMINISTRATOR
3                    true    false      false      NO ACCESS
4                    true    false      false      NO ACCESS
5                    true    false      false      NO ACCESS
6                    true    false      false      NO ACCESS
7                    true    false      false      NO ACCESS
8                    true    false      false      NO ACCESS
9                    true    false      false      NO ACCESS
10                   true    false      false      NO ACCESS
```

# 11.6. Inband Management

## 11.6.1. In-band Management Overview

RBFS is mostly deployed on the RtBrick Host Operating System as a Linux container. The RtBrick Host is only reachable through the out-of-band management interface. In order to use services like NTP, and TACACS, which are run on RtBrick Host, or to use services like ssh, snmpd running in LXC container, one must use an out-of-band management connection. In-band management provides a way to access these services which are running in RtBrick Host and LXC containers via physical traffic ports.

The RBFS creates a Linux kernel interface named `inband-mgmt-0` when in-band management is enabled on an instance. The loopback IPs of the in-band instance are then assigned to this Linux interface, and the routes of this instance are downloaded to the LXC container, then to RtBrick Host. Trap rules are installed in the hardware depending on the in-band service enabled.

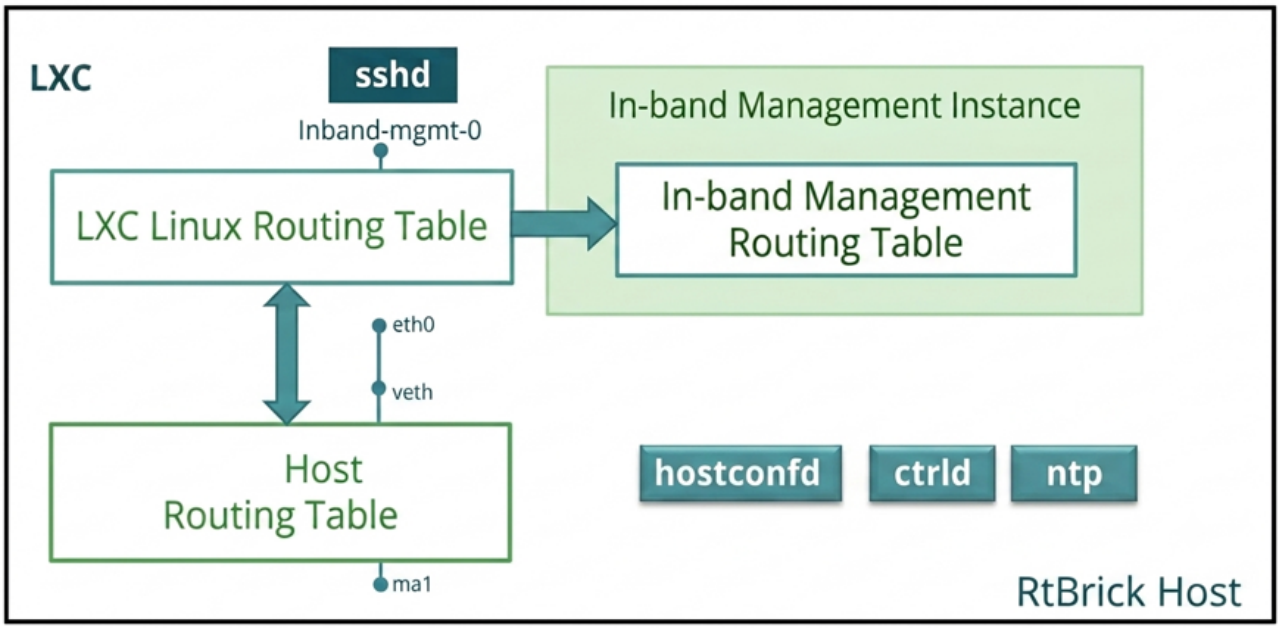



*Figure 172. In-band Management Overview*

#### Support for Non-Loopback Interfaces

In-band management is supported on non-loopback interfaces. It allows you to use the non-loopback interfaces for management protocols such as SSH. Supported management protocols can connect to and from physical interface addresses, enabling physical data interfaces to be used as primary or secondary management interfaces.

In the configuration, when `loopback-only` is set to `true`, management protocols such as SSH, CTRLD, and other management services can be accessed only through loopback interfaces that belong to the in-band management instance. Non-loopback interface addresses are not used for management access.

#### Supported Platforms

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

### 11.6.2. In-band Management Configuration

#### Enabling In-band Management in an Instance

Use the following CLI syntax to enable in-band management in an instance:

**set inband-management instance** <instance-name>

| Attribute | Description |
|---|---|
| `<instance-name>` | Routing instance name in which in-band management has to be enabled. All the IFLs in this instance will be enabled with in-band management service after executing this command |

The following example configures the `management` instance in which in-band management will be enabled:

```
set inband-management instance management
```

The following example shows in-band management in an instance:

```
    "rtbrick-config:inband-management": {
      "instance": [
        {
          "name": "management",
        }
      ]
    }
```

API To access the RESTCONF API that corresponds to this CLI, click here.

#### Enabling In-band Management Services

**Syntax:**

**set inband-management instance** <instance-name> <service> <true/false>

| Attribute | Description ` |
|---|---|
| <instance-name>` | Routing instance name in which in-band management has to be enabled. All the IFLs in this instance will be enabled with in-band management service after executing this command |
| `<service>` | Specifies the supported services to enable: apigw, ctrld, ntp, snmp, ssh, tacacs, telnet. By enabling any of these services, hosts reachable via the physical interface in the inband instance can access the services. |
| `<true \| false>` | A `true` value enables the service. A `false` value disables the service. |

Example: Enabling In-band Management Services

```
"rtbrick-config:inband-management": {
  "instance": [
    {
      "name": "management",
      "ssh": "true",
      "ctrld": "true"
    }
  ]
}
```

##### Enabling API Gateway (APIGW) Service

To access the APIGW service running in the RtBrick Host, this service has to be enabled in in-band management.

**Syntax:**

**set inband-management instance** <instance-name> **apigw** <true/false>

| Attribute | Description |
|---|---|
| `<instance-name>` | Routing instance name in which in-band management has to be enabled. |
| `<true \| false>` | A `true` value enables the APIGW service. A `false` value disables the APIGW service. |

Example: Enabling APIGW In-band Management Services

```
"rtbrick-config:inband-management": {
  "instance": [
    {
      "name": "management",
      "apigw": "true",
    }
  ]
}
```

#### Enabling CTRLD Service

To access the CTRLD service running in the RtBrick Host, the CTRLD service has to be enabled in in-band management.

**Syntax:**

**set inband-management instance** <instance-name> **ctrld** <true/false>

| Attribute | Description |
|---|---|
| `<instance-name>` | Routing instance name in which in-band management has to be enabled. |
| `<true \| false>` | A `true` value enables the CTRLD service. A `false` value disables the CTRLD service. |

Example: Enabling CTRLD In-band Management Services

```
"rtbrick-config:inband-management": {
  "instance": [
    {
      "name": "management",
      "ctrld": "true",
    }
  ]
}
```

#### Enabling NTP service

To access the NTP service running in the RtBrick Host, this service has to be enabled in in-band management.

**Syntax:**

**set inband-management instance** <instance-name> **ntp** <true/false>

| Attribute | Description |
|---|---|
| `<instance-name>` | Routing instance name in which in-band management has to be enabled. |
| `<true \| false>` | A `true` value enables the ntp service. A `false` value disables the ntp service. |

Example: Enabling NTP In-band Management Services

```
"rtbrick-config:inband-management": {
  "instance": [
    {
      "name": "management",
      "ntp": "true",
    }
  ]
}
```

#### Enabling SNMP service

To access the Simple Network Management Protocol (SNMP) service running in the RtBrick Host, this service has to be enabled in in-band management.

**Syntax:**

**set inband-management instance** <instance-name> **snmp** <true/false>

| Attribute | Description |
|---|---|
| `<instance-name>` | routing instance name in which in-band management has to be enabled. |
| `<true \| false>` | A `true` value enables the SNMP service. A `false` value disables the SNMP service. |

Example: Enabling SNMP In-band Management Services

```
"rtbrick-config:inband-management": {
  "instance": [
    {
      "name": "management",
      "snmp": "true",
    }
  ]
}
```

#### Enabling SSH service

To access the ssh service running in the LXC container hosting RBFS, ssh service has to be enabled.

**Syntax:**

**set inband-management instance** <instance-name> **ssh** <true/false>

| Attribute | Description |
|---|---|
| `<instance-name>` | Routing instance name in which in-band management has to be enabled. |
| `<true \| false>` | A 'true' value enables the ssh service. A 'false' value disables the ssh service. |

Example: Enabling SSH In-band Management Services

```
"rtbrick-config:inband-management": {
  "instance": [
    {
      "name": "management",
      "ssh": "true",
    }
  ]
}
```

#### Enabling TACACS Service

To access the TACACS service running in the RtBrick Host, this service has to be enabled in in-band management.

**Syntax:**

**set inband-management instance** <instance-name> **tacacs** <true/false>

| Attribute | Description |
|---|---|
| `<instance-name>` | Routing instance name in which in-band management has to be enabled. |
| `<true \| false>` | A `true` value enables the TACACS service. A `false` value disables the TACACS service. |

Example: Enabling TACACS In-band Management Services

```
"rtbrick-config:inband-management": {
  "instance": [
```

```
        {
          "name": "management",
          "tacacs": "true",
        }
      ]
    }
```

#### Enabling Telnet Service

To access the telnet service running in the LXC container hosting RBFS, telnet service has to be enabled.

**Syntax:**

**set inband-management instance** <instance-name> **telnet** <true/false>

| Attribute | Description |
|---|---|
| `<instance-name>` | Routing instance name in which in-band management has to be enabled. |
| `<true \| false>` | A 'true' value enables the telnet service. A 'false' value disables the telnet service. |

Example: Enabling Telnet In-band Management Services

```
    "rtbrick-config:inband-management": {
      "instance": [
        {
          "name": "management",
          "telnet": "true",
        }
      ]
    }
```

#### Enabling Connection Tracking

Enabling connection tracking in inband installs dynamic ACLs for all the connection/packet initiated by the device so that the response packets are not dropped at the hardware.

**Syntax:**

**set inband-management instance** <instance-name> **connection-tracking true**

| Attribute | Description |
| --- | --- |
| `<instance-name>` | Routing instance name in which in-band management has to be enabled. |
| `true` | Enables all in-band management services. |

Example: Enabling Connection Tracking in In-band Management

```
"rtbrick-config:inband-management": {
    "instance": [
      {
        "name": "default",
        "connection-tracking": "true"
      }
    ]
  }
```

#### Enabling LIX1 Service

To enable or disable the "lix1" service within inband management, use the following CLI command:

**set inband-management instance** <instance-name> **lix1** <true | false>

| Attribute | Description |
| --- | --- |
| `<instance-name>` | Specifies the routing instance where the in-band lix1 service is to be enabled. |
| `<true \| false>` | Setting this attribute to "true" enables the "lix1" service, while setting it to "false" disables the service. |

Example: Enabling the "lix1" service in inband management

```
    "rtbrick-config:inband-management": {
      "instance": [
        {
          "name": "default",
          "lix1": "true",
        }
      ]
    }
```

#### Enabling All Services in In-band Management

Enabling this service will allow access to all services running in LXC/RtBrick Host. Once this service is enabled, packets that don’t hit any of the other acls/services in RBFS are redirected to LXC/RtBrick Host.

**Syntax:**

**set inband-management instance** <instance-name> **all true**

| Attribute | Description |
|---|---|
| `<instance-name>` | Routing instance name in which in-band management has to be enabled. |
| `true` | Enables all in-band management services. |

Example: Enabling all In-band Management Services

```
"rtbrick-config:inband-management": {
  "instance": [
    {
      "name": "management",
      "all": "true",
    }
  ]
}
```

#### Enabling In-band Management for a Specific Source

You can enable in-band management services (as mentioned in the previous sections). It will expose this service to all the sources which are reachable via in-band service.

The in-band management `source-prefix-list` option supports various prefix ranges such as /16, /24, and /32. It can support a prefix list with any IPv4 prefixes.

To restrict this to specific source prefixes, source-prefix-list has to be enabled using the following command.

By configuring this, the hosts having IPs in the mentioned source prefix list only can access this service.

**Syntax:**

**set inband-management instance** <instance-name> **source-prefix-list** <source-prefix-list-name>

| Attribute | Description |
|---|---|
| `<instance-name>` | Routing instance name in which in-band management has to be enabled. |

| Attribute | Description |
| --- | --- |
| `<source-prefix-list-name>` | Specifies the name of the source prefix-list which is configured with `set forwarding-options prefix-list` command. For details of configuring prefix-list, refer to Configuring Prefix Lists section in the ACL User Guide. |

Example: Enabling source prefix list in In-band Management Services

```
    "rtbrick-config:inband-management": {
      "instance": [
        {
          "name": "management",
          "source-prefix-list": "source-prefix1"
        }
      ]
    }
```

The following example configuration enables multiple prefix lists for in-band management.

```
set inband-management instance inband-vrf ssh true
set inband-management instance inband-vrf source-prefix-list rtbrick-p1
set forwarding-options prefix-list rtbrick-p1 ipv4-prefix 198.51.100.0/24
set forwarding-options prefix-list rtbrick-p1 ipv4-prefix 203.0.113.0/24
set forwarding-options prefix-list rtbrick-p1 ipv4-prefix 192.0.2.1/32
set instance inband-vrf address-family ipv4 unicast
set instance inband-vrf address-family ipv6 unicast
```

```
"rtbrick-config:inband-management": {
      "instance": [
        {
          "name": "inband-vrf",
          "ssh": "true",
          "source-prefix-list": "rtbrick-p1"
        }
      ]
    },
    "rtbrick-config:forwarding-options": {
      "prefix-list": [
        {
          "prefix-list-name": "rtbrick-p1",
          "ipv4-prefix": [
            {
              "ipv4-prefix": "198.51.100.1/24"
            },
            {
              "ipv4-prefix": "203.0.113.0/24"
            },
```

```
            {
              "ipv4-prefix": "192.0.2.1/32"
            }
          ]
        }
      ]
    },
    "rtbrick-config:instance": [
      {
        "name": "inband-vrf",
        "address-family": [
          {
            "afi": "ipv4",
            "safi": "unicast"
          },
          {
            "afi": "ipv6",
            "safi": "unicast"
          }
        ]
      }
    ],
```

#### Setting Loopback-Only Interfaces of SSH Service

By default, you both loopback and non-loopback interfaces are available for In-band Management services. When `loopback-only` is set to `true`, management protocols such as SSH, CTRLD, and other management services can be accessed only through loopback interfaces that belong to the in-band management instance. Non-loopback interface addresses are not used for management access.

**Syntax:**

**set inband-management instance** <name> **loopback-only true**

The following table provide the command options and descriptions.

| Attribute | Description |
|---|---|
| `instance <name>` | Name of the instance. |
| `loopback-only true` | Inband management access is available only through loopback interfaces that belong to the in-band management instance. |

The following example configures the in-band management instance 'default' to allow management access only through loopback interface addresses. When `loopback-only` is set to `true`, non-loopback interface addresses are excluded from in-band management access.

```
set inband-management instance default loopback-only true
```

```
{
  "rtbrick-config:inband-management": {
    "instance": [
      {
        "name": "default",
        "loopback-only": "true"
      }
    ]
  }
}
```

### 11.6.3. In-Band Management Operational Commands

The In-band Management show commands provide detailed information about the In-band Management operations.

#### Verifying In-band Management on LXC Container

In the LXC container, there will be a new interface named `inband-mgmt-0` on enabling in-band management. All the loopback address as well as route in in-band instance should be assigned to this interface.

The example below shows how to verify if `inband-mgmt-0` interface is created. Using the following command check if `inband-mgmt-0` interface is created.

```
supervisor@rtbrick:~$ ip link show
<...>
5: inband-mgmt-0: <POINTOPOINT,UP,LOWER_UP> mtu 1492 qdisc pfifo_fast state UNKNOWN
mode DEFAULT group default qlen 500
    link/none
```

Now verify that all the loopback addresses in the in-band management instance are added to Linux:

```
3: inband-mgmt-0: <POINTOPOINT,UP,LOWER_UP> mtu 1492 qdisc pfifo_fast state UNKNOWN
group default qlen 500
    link/none
    inet 192.0.2.1/32 scope global inband-mgmt-0
       valid_lft forever preferred_lft forever
    inet 198.51.100.1/32 scope global inband-mgmt-0
       valid_lft forever preferred_lft forever
    inet6 fe80::d5c5:38ad:93c:308b/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
<...>
```

Verify all the routes in the same instance are also downloaded:

```
supervisor@rtbrick:~ $ ip route show | grep 'inband'
192.0.2.0/30 dev inband-mgmt-0 proto rtb_fibd scope link
192.0.2.1 dev inband-mgmt-0 proto rtb_fibd scope link
192.0.2.4/30 dev inband-mgmt-0 proto rtb_fibd scope link
192.0.2.5 dev inband-mgmt-0 proto rtb_fibd scope link
192.0.2.100/30 dev inband-mgmt-0 proto rtb_fibd scope link
192.0.2.101 dev inband-mgmt-0 proto rtb_fibd scope link
192.0.2.104/30 dev inband-mgmt-0 proto rtb_fibd scope link
192.0.2.105 dev inband-mgmt-0 proto rtb_fibd scope link
198.51.100.201 dev inband-mgmt-0 proto rtb_fibd scope link
198.51.100.202 dev inband-mgmt-0 proto rtb_fibd scope link
```

**Verifying In-band Management on RtBrick Host**

The example below shows how to verify Linux routing tables on the Host Operating System and an LXC container. The same routes downloaded into the LXC container should also be present in RtBrick Host.

```
supervisor@rtbrick:~ $ ip route show
default via 10.0.3.1 dev lxc-mgmt-0 proto rtb_routesync scope rtb_umc metric
4294966400
10.0.3.0/24 dev lxc-mgmt-0 proto kernel scope link src 10.0.3.10
10.200.138.0/25 via 10.0.3.1 dev lxc-mgmt-0 proto rtb_routesync scope rtb_umc metric
128
10.200.138.34 via 10.0.3.1 dev lxc-mgmt-0 proto rtb_routesync scope rtb_umc metric 128
192.0.2.0/30 dev inband-mgmt-0 proto rtb_fibd scope link
192.0.2.1 dev inband-mgmt-0 proto rtb_fibd scope link
192.0.2.4/30 dev inband-mgmt-0 proto rtb_fibd scope link
192.0.2.5 dev inband-mgmt-0 proto rtb_fibd scope link
192.0.2.100/30 dev inband-mgmt-0 proto rtb_fibd scope link
192.0.2.101 dev inband-mgmt-0 proto rtb_fibd scope link
192.0.2.104/30 dev inband-mgmt-0 proto rtb_fibd scope link
192.0.2.105 dev inband-mgmt-0 proto rtb_fibd scope link
198.51.100.201 dev inband-mgmt-0 proto rtb_fibd scope link
198.51.100.202 dev inband-mgmt-0 proto rtb_fibd scope link
```

# 11.7. DHCP Relay/Proxy

## 11.7.1. DHCP Relay Overview

The Dynamic Host Configuration Protocol (DHCP) is a standardized client/server network protocol that dynamically assigns IPv4 addresses and other related configuration information to network devices.

DHCP provides an automated way to distribute and update IPv4 addresses and other configuration information on a network. A DHCP server provides this information to a DHCP client through the exchange of a series of messages, known as the DHCP conversation or the DHCP transaction. If the

DHCP server and DHCP clients are located on different subnets, a DHCP relay agent is used to facilitate the conversation.

The RBFS DHCP relay agent handles all DHCP packets received on logical interfaces with the relay agent functionality enabled. This includes all DHCP broadcast and unicast packets to ensure that the whole DHCP communication between client and server can be tracked for operational and security reasons. The DHCP packets received from clients will than be forwarded to one or more DHCP servers to support high availability.

All DHCP packets send from the relay agent to the configured DHCP servers will be send with the IP address of the gateway interface (giaddr) or a dedicated source address per server. This allows to better traverse firewalls between relay agent and server.

**Supported Platforms**

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

**DHCP Relay Agent**

The following are the operation steps involved by the DHCP relay agent with multiple clients and servers:

1. DHCP client sends DISCOVER packets onto the LAN.
2. DHCP relay agent receives the DISCOVER packet, creates a client binding and forwards it to the configured server(s) in different subnets.
3. DHCP server replies with OFFER packet.
4. DHCP relay agent on receiving the OFFER, sends it to the client based on the binding.
5. DHCP client sends a REQUEST after receiving the OFFER.
6. DHCP relay agent forwards the REQUEST to the configured server(s).
7. Selected DHCP server replies to the REQUEST with an ACK.
8. DHCP relay agent receives the ACK and forward it to client.
9. Client will start using the IP and configuration after receiving the ACK.
10. DHCP relay agent listen to the unicast communication between the client and server and update the bindings.

**DHCP Relay in Proxy Mode**

The proxy mode is an enhancement for the DHCP relay function to hide and protect the actual DHCP servers from the clients. For servers this mode is transparent but for clients it seems that the relay agent is the server and the actual servers are not visible anymore.

The following are the operation steps involved by the DHCP relay agent in proxy mode with multiple clients and servers:

1. DHCP client sends DISCOVER packets onto the LAN.

2. DHCP relay agent receives the DISCOVER packet, creates a client binding and forwards it to the configured server(s) in different subnets.
3. DHCP server replies with OFFER packet.
4. DHCP relay agent on receiving OFFERs from multiple servers, select the OFFER from the first server, replaces the server identifier with relay agent address, update the binding and sends it to the client.
5. DHCP client sends a REQUEST to relay agent address after receiving the OFFER.
6. DHCP relay agent forwards the REQUEST to the configured server(s).
7. Selected DHCP server replies to the REQUEST with an ACK.
8. DHCP relay agent receives the ACK, replaces the server identifier with relay agent address and forward it to client.
9. Client will start using the IP and configuration after receiving the ACK.
10. Client request the relay agent for the lease renewals, which relay agent will forward to the selected server.

In all the steps above, the client remains unaware of the actual DHCP server.

### 11.7.2. DHCP Relay Configuration

DHCP relay requires a pre-provisioned logical interface with at least an local or borrowed (unnumbered) IPv4 address configured which is than used a gateway IP address (giaddr) by the DHCP relay function.

There are two steps required to enable the DHCP relay functionality:

1. DHCP server configuration
2. DHCP relay configuration

**DHCP Server Configuration**

This configuration defines the DHCP servers which are referenced by the actual DHCP relay configuration.

```
supervisor@rtbrick>LEAF01: cfg> set access dhcp-server <server>
  <cr>
  address                DHCP server address
  routing-instance       Instance name from which DHCP server is reachable
  source-address         Source address used for DHCP packets
```

The following example shows a typical DHCP server configuration.

```
{
  "rtbrick-config:access": {
    "dhcp-server": [
      {
```

```
        "server-name": "server1",
        "address": "198.51.100.101",
        "routing-instance": "default"
      }
    ]
  }
}
```

| Attribute | Description |
|---|---|
| `server` | Specifies the DHCP server name. |
| `address` | Specifies the IPv4 address of the DHCP server. |
| `routing-instance` | Routing instance from which DHCP server is reachable.<br><br>**Default:** default |
| `source-address <source-address>` | Specifies the source IPv4 address to be used to reach DHCP server.<br><br>**Default:** gateway interfaces address (giaddr) |

#### DHCP Relay Configuration

This configuration enables the DHCP relay function on the corresponding logical interface (IFL).

```
supervisor@rtbrick>LEAF01: cfg> set access dhcp-relay <interface-name>
  <cr>
  agent-circuit-id      Add Agent-Circuit-Id (option 82)
  agent-remote-id       Add Agent-Remote-ID (option 82)
  dhcp-server           DHCP server
  proxy-mode            Enable relay proxy mode
```

The following example shows a typical DHCP relay configuration.

```
{
    "rtbrick-config:access": {
      "dhcp-relay": [
        {
          "interface": "ifl-0/0/1/1",
          "dhcp-server": [
            "server1",
            "server2"
           ]
        }
      ]
    }
  }
```

| Attribute | Description |
|---|---|
| `interface-name` | Logical interface name on which client packets are expected. |
| `dhcp-server <dhcp-server>` | DHCP server name to which client packets has to be forwarded (multiple servers can be configured). |
| `proxy-mode` | Enable the relay proxy mode for this interface. |
| `agent-circuit-id <aci>` | Agent-Circuit-Id to be added in DHCP option 82. |
| `agent-remote-id <ari>` | Agent-Remote-Id to be added in DHCP option 82. |

#### DHCP Relay on Unnumbered Interface

DHCP Relay is supported over unnumbered interfaces. The configuration is similar to the regular DHCP relay configuration. The only change is that the logical interface will have a borrowed IP address.

The following example shows a typical DHCP relay configuration over unnumbered interface.

```
{
  "data": {
    "rtbrick-config:interface": [
      {
        "name": "ifp-0/0/1",
        "unit": [
          {
            "unit-id": 1,
            "unnumbered": {
              "interface": "lo-0/0/1/1"
            }
          }
        ]
      },
      {
        "name": "ifp-0/0/2",
        "unit": [
          {
            "unit-id": 1,
            "unnumbered": {
              "interface": "lo-0/0/1/1"
            }
          }
        ]
      },
      {
        "name": "lo-0/0/1",
        "unit": [
          {
            "unit-id": 1,
            "address": {
              "ipv4": [
```

```
                {
                  "prefix4": "198.51.100.71/24"
                }
              ]
            }
          }
        ]
      }
    ],
    "rtbrick-config:access": {
      "dhcp-relay": [
        {
          "interface": "ifl-0/0/1/1",
          "dhcp-server": [
            "server1",
            "server2"
            ]
        },
        {
          "interface": "ifl-0/0/2/1",
          "dhcp-server": [
            "server1",
            "server2"
            ]
        }
      ]
    }
  }
}
```

### 11.7.3. DHCP Relay Operational Commands

#### Verify DHCP Relay Configuration

A good starting point for any troubleshooting is to get an overview about the configuration. First check the DHCP server configuration and optionally check the reachability using additional commands like `ping`.

```
supervisor@rtbrick>LEAF01: cfg> show config access dhcp-server
{
    "rtbrick-config:dhcp-server": [
      {
        "server-name": "server1",
        "address": "198.51.100.15",
        "source-address": "198.51.100.101",
        "routing-instance": "default"
      }
    ]
  }
```

In the next step it should be verified that the DHCP relay function is enabled for the desired logical interfaces. Those interfaces must be configured with a valid local or borrowed (unnumbered) IPv4 address.

```
supervisor@rtbrick>LEAF01: cfg> show config access dhcp-relay
{
    "rtbrick-config:dhcp-relay": [
      {
        "interface": "ifl-0/0/1/1",
        "dhcp-server": [
          "server1",
          "server2"
          ]
      }
    ]
  }
```

**Verify DHCP Relay Packet Processing**

All DHCP packets processed by RBFS will be handled by the IPoE daemon (`rtbrick-ipoed.1`) which separates between DHCP relay and subscriber packets. The command `show dhcp statistics` gives some statistics about the DHCP packets send and received by the IPoE daemon.

Example: DHCP statistics

```
supervisor@rtbrick: op> show dhcp statistics
Global DHCPv4 statistics
Total packets received      : 1
Decode error                : 0

IP DHCPv4 relay statistics
Relay packets received      : 1
Relay packets sent          : 1
Relay encode error          : 0
Relay send error            : 0

IPoE DHCPv4 subscriber statistics
IPoE packets received       : 0
IPoE packets sent           : 0
IPoE encode error           : 0
IPoE send error             : 0
```

Received malformed packets will be counted here as part of the shared DHCP packet infrastructure.

Those statistics can be reset with the command `clear dhcp statistics`.

In the next step, the command `show dhcp relay statistics` provides more detailed statistics for those packets handled by the DHCP relay function.

Example: DHCP relay statistics

```
supervisor@rtbrick: op> show dhcp relay statistics
Packet           Received         Sent
DHCP DISCOVER  : 2                2
DHCP OFFER     : 1                1
DHCP REQUEST   : 1                1
DHCP DECLINE   : 0                0
DHCP ACK       : 1                1
DHCP NAK       : 0                0
DHCP RELEASE   : 0                0
DHCP INFORM    : 0                0

Errors:

Invalid client packets received : 0
Invalid server packets received : 0
I/O errors                      : 0
Configuration errors            : 0
Dropped min elapsed             : 0
```

```
supervisor@rtbrick: op> show dhcp relay statistics
Packet           Received         Sent
DHCP DISCOVER  : 2                2
DHCP OFFER     : 1                1
DHCP REQUEST   : 1                1
DHCP DECLINE   : 0                0
DHCP ACK       : 1                1
DHCP NAK       : 0                0
DHCP RELEASE   : 0                0
DHCP INFORM    : 0                0

Errors:

Invalid client packets received : 0
Invalid server packets received : 0
I/O errors                      : 0
Configuration errors            : 0
Dropped min elapsed             : 0
```

A DHCP packet is consider as invalid if correctly formed but some invalid or unexpected values like a DHCP hop count greater than 16, messages from server without server identifier option, non-matching transaction (XID) and many more.

Those statistics can be reset with the command `clear dhcp relay statistics`.

### Verify DHCP Relay Bindings

The DHCP relay function tracks all active DHCP communications as so called bindings. The command `show dhcp relay binding` allows to verify those bindings with extensive filters to better navigate in scaled environments with thousands of bindings.

**Syntax**:

**show dhcp relay binding** <option>

| Option | Description |
|---|---|
| `-` | Without any option, the commands displays all DHCP relay binding details. |
| `detail` | Detailed binding information. |
| `filter` | Filter DHCP relay binding. |
| `interface` | Binding information for the specified interface. |

#### DHCP Relay Binding

**Syntax**:

**show dhcp relay binding filter** <option>

| Option | Description |
|---|---|
| `address` | Filters DHCP relay bindings based on the specified client IP address. |
| `count` | Displays only the count of bindings that match the applied filter criteria. |
| `detail` | Displays detailed information for DHCP relay bindings that match the filter. |
| `interface` | Filters bindings associated with a specific interface. |
| `interface-regex` | Filters bindings for interfaces that match the specified regular expression. |
| `mac` | Filters bindings based on the specified client hardware (MAC) address. |
| `mac-regex` | Filters bindings for MAC addresses that match the specified regular expression. |
| `server` | Filters bindings associated with a specific DHCP server identifier. |

Example: DHCP relay binding details

```
supervisor@rtbrick: op> show dhcp relay binding detail
Interface                  : ifl-0/1/23/100
Hardware Address           : 02:00:00:00:00:01
Client IP Address          : 203.0.113.200
Routing Instance           : default
Server Identifier          : 192.0.2.2
Gateway IP Address         : 203.0.113.129
Lease Time                 : 35.0 sec
Lease Start                : Fri Jun 05 04:45:50 GMT +0000 2026
Lease Expires              : 29.0 sec
Last Packet Received       : Fri Jun 05 04:45:50 GMT +0000 2026
Last Received Packet Type  : ACK
Last Packet Sent           : Fri Jun 05 04:45:50 GMT +0000 2026
Last Sent Packet Type      : ACK
```

The most extensive output is provided by selecting a particular binding using the interface and client MAC address as key.

Example: DHCP relay binding interface information

```
supervisor@rtbrick: op> show dhcp relay binding interface ifl-0/1/23/100 mac
02:00:00:00:00:01
Interface                  : ifl-0/1/23/100
Hardware Address           : 02:00:00:00:00:01
Client IP Address          : 203.0.113.200
Routing Instance           : default
Server Identifier          : 192.0.2.2
Gateway IP Address         : 203.0.113.129
Lease Time                 : 35.0 sec
Lease Start                : Fri Jun 05 04:45:50 GMT +0000 2026
Lease Expires              : 30.0 sec
Last Packet Received       : Fri Jun 05 04:45:50 GMT +0000 2026
Last Received Packet Type  : ACK
Last Packet Sent           : Fri Jun 05 04:45:50 GMT +0000 2026
Last Sent Packet Type      : ACK

Statistics:

Packet           Received         Sent
DHCP DISCOVER  : 1                1
DHCP OFFER     : 1                1
DHCP REQUEST   : 1                1
DHCP DECLINE   : 0                0
DHCP ACK       : 1                1
DHCP NAK       : 0                0
DHCP RELEASE   : 0                0
DHCP INFORM    : 0                0

Errors:
```

```
Invalid client packets received : 0
Invalid server packets received : 0
I/O errors                      : 0
Configuration errors            : 0
Dropped min elapsed             : 0
```

This output returns additional statistics per DHCP relay binding.

#### Clear Commands

The command `clear dhcp relay binding` removes either all bindings, all bindings of a corresponding interface or only one particular binding.

**Syntax**:

**clear dhcp relay binding**

| Option | Description |
|---|---|
| `all` | Clears all DHCP relay bindings from the system. |
| `interface` | Clears DHCP relay bindings associated with the specified interface. |

Example:

```
clear dhcp relay binding all
clear dhcp relay binding interface ifp-0/0/1/3
clear dhcp relay binding interface ifp-0/0/1/3 mac 02:00:00:00:00:01
```

| Registered Address | Support | Sales |
| --- | --- | --- |
| 40268, Dolerita Avenue Fremont CA 94539 | | |
| +1-650-351-2251 | | +91 80 4850 5445 |
| http://www.rtbrick.com | support@rtbrick.com | sales@rtbrick.com |


©Copyright 2024 RtBrick, Inc. All rights reserved. The information contained herein is subject to change without notice. The trademarks, logos and service marks ("Marks") displayed in this documentation are the property of RtBrick in the United States and other countries. Use of the Marks are subject to RtBrick's Term of Use Policy, available at https://www.rtbrick.com/privacy. Use of marks belonging to other parties is for informational purposes only.