

Technical Documentation

Reference Designs

Version 26.2.1.5, 18 November 2026

Table of Contents

1. RBFS Multiservice Edge for PPPoE Subscribers . . . 2
   1.1. An Overview of RBFS Multiservice Edge for PPPoE Subscribers . . . 2
      1.1.1. About RBFS Multiservice Edge . . . 2
      1.1.2. Deployment . . . 4
      1.1.3. Using the RBFS CLI . . . 4
   1.2. Configuration and Settings . . . 5
      1.2.1. Platform Configuration and Settings . . . 5
      1.2.2. General Configuration . . . 6
   1.3. Protocol Configurations . . . 19
      1.3.1. Configure IS-IS . . . 19
      1.3.2. Configure LDP on the Interfaces . . . 20
      1.3.3. Configure BGP . . . 21
      1.3.4. Configuring the Service Node . . . 25
      1.3.5. Validating Protocols on RBFS Multiservice Edge . . . 28
   1.4. PPPoE Subscriber Management Configuration . . . 31
      1.4.1. Configuring PPPoE Subscriber Management . . . 32
      1.4.2. PPPoE Quality of Service (QoS) Configuration . . . 38
      1.4.3. Configure Mirroring for Downstream Traffic Remark Validation . . . 57
      1.4.4. Configure FreeRADIUS Server . . . 59
      1.4.5. Validating PPPoE Subscriber Bring-Up . . . 64
      1.4.6. PPPoE Subscriber Accounting for Upstream and Downstream Traffic . . . 68
      1.4.7. Configuring Lawful Intercept (LI) . . . 69
   1.5. Appendixes . . . 69
      Appendix A: RBFS Multiservice Edge Configuration . . . 69
      Appendix B: TACACS+ Server Configuration . . . 69
      Appendix C: RADIUS Server Configuration . . . 69
      Appendix D: BNG Blaster Configuration . . . 70
2. RBFS Multiservice Edge for IPoE Subscribers . . . 71
   2.1. An Overview of RBFS Multiservice Edge for IPoE Subscribers . . . 71
      2.1.1. About the RBFS Multiservice Edge . . . 71
      2.1.2. Deployment . . . 73
      2.1.3. Using the RBFS CLI . . . 73
   2.2. Configuration and Settings . . . 74
      2.2.1. Platform Configuration and Settings . . . 74
      2.2.2. General Configuration . . . 75
   2.3. Protocol Configurations . . . 88
      2.3.1. Configure OSPF . . . 88
      2.3.2. Configure LDP on the Interfaces . . . 89

2.3.3. Configure BGP . . . 90
2.3.4. Configuring the Service Node . . . 93
2.3.5. Validating Protocols on RBFS Multiservice Edge . . . 96
2.4. IPoE Subscriber Management Configuration . . . 100
2.4.1. Configuring IPoE Subscriber Management . . . 100
2.4.2. IPoE Quality of Service (QoS) Configuration . . . 106
2.4.3. Configure Mirroring for Downstream Traffic Remark Validation . . . 120
2.4.4. Configure FreeRADIUS Server . . . 121
2.4.5. Validating IPoE Subscriber Bring-Up . . . 127
2.4.6. IPoE Subscriber Accounting for Upstream and Downstream Traffic . . . 131
2.4.7. Configuring Lawful Intercept (LI) . . . 131
2.5. Appendixes . . . 131
Appendix A: RBFS Multiservice Edge Configuration . . . 132
Appendix B: TACACS+ Server Configuration . . . 132
Appendix C: RADIUS Server Configuration . . . 132
Appendix D: BNG Blaster Configuration . . . 132
3. RBFS Multiservice Edge with CGNAT for IPoE and PPPoE Subscribers . . . 133
3.1. RBFS Multiservice Edge with CGNAT for IPoE and PPPoE Subscribers . . . 133
3.1.1. Overview . . . 133
3.1.2. About RBFS Multiservice Edge . . . 133
3.1.3. About BNG Blaster . . . 134
3.1.4. About RBFS Carrier-Grade NAT . . . 134
3.1.5. RBFS Multiservice Edge with CGNAT Implementation Architecture . . . 134
3.1.6. Deploying RBFS Multiservice Edge . . . 136
3.1.7. Using the RBFS CLI . . . 136
3.2. Configuration and Settings . . . 137
3.2.1. Platform Configuration and Settings . . . 137
3.2.2. General Configuration . . . 138
3.3. Protocol Configurations . . . 152
3.3.1. Configure OSPFv2/v3 . . . 152
3.3.2. Configure LDP on the Interfaces . . . 154
3.3.3. Configure BGP . . . 155
3.3.4. Configuring the Service Node . . . 158
3.3.5. Validating Protocols on RBFS Multiservice Edge . . . 162
3.4. RBFS Subscriber Management Configuration . . . 166
3.4.1. PPPoE Subscriber Management Configuration . . . 166
3.4.2. IPoE Subscriber Management Configuration . . . 174
3.4.3. Quality of Service (QoS) Configuration . . . 180
3.4.4. Configure FreeRADIUS Server . . . 191
3.5. CGNAT Configuration . . . 197
3.5.1. Configure CGNAT . . . 197

3.6. Appendixes . . . 201
Appendix A: RBFS Multiservice Edge Configuration . . . 201
Appendix B: TACACS+ Server Configuration . . . 201
Appendix C: RADIUS Server Configuration . . . 201
Appendix D: BNG Blaster Configuration . . . 201
4. RBFS Multiservice Edge with Redundancy for IPoE Subscribers . . . 202
4.1. RBFS Multiservice Edge with Redundancy for IPoE Subscribers . . . 202
4.1.1. About this Guide . . . 202
4.1.2. About the RBFS Multiservice Edge . . . 202
4.1.3. Deployment . . . 202
4.1.4. Using the RBFS CLI . . . 203
4.1.5. About RBFS Redundancy . . . 203
4.1.6. Redundancy Pair BNG Devices and Inter-BNG Connectivity . . . 204
4.1.7. RBFS Multiservice Edge with Redundancy for IPoE Subscribers Implementation Architecture . . . 204
4.2. Configuration and Settings . . . 206
4.2.1. Platform Configuration and Settings . . . 206
4.2.2. General Configuration . . . 207
4.3. Protocol Configurations . . . 221
4.3.1. Configure IS-IS . . . 221
4.3.2. Configure LDP on the Interfaces . . . 222
4.3.3. Configure BGP . . . 223
4.3.4. Configuring the Service Node . . . 227
4.3.5. Validating Protocols on RBFS Multiservice Edge . . . 229
4.4. IPoE Subscriber Management Configuration . . . 233
4.4.1. Configuring IPoE Subscriber Management . . . 234
4.4.2. IPoE Quality of Service (QoS) Configuration . . . 240
4.4.3. Configure Mirroring for Downstream Traffic Remark Validation . . . 258
4.4.4. Configure FreeRADIUS Server . . . 260
4.4.5. Validating IPoE Subscriber Bring-Up . . . 266
4.4.6. IPoE Subscriber Accounting for Upstream and Downstream Traffic . . . 269
4.5. RBFS Redundancy Configurations . . . 270
4.5.1. Configure RBFS Redundancy . . . 270
4.5.2. BNG Blaster Configuration . . . 284
4.5.3. Validating Multiservice Edge Redundancy and Reachability . . . 289
4.6. Appendixes . . . 293
Appendix B: TACACS+ Server Configuration . . . 293
Appendix B: RADIUS Server Configuration . . . 293
Appendix C: RBFS Multiservice Edge1 Configuration . . . 293
Appendix D: RBFS Multiservice Edge2 Configuration . . . 293
Appendix E: R-1 (Router 1) Configuration . . . 293

Appendix F: R-2 (Router 2) Configuration 293
Appendix G: Edge Router Configuration 293
Appendix H: BNG Blaster Configurations 293

RBFS Multiservice Edge Reference Designs offer tested solutions for setting up devices for lab tests. Each guide presents a complete use case with topology, platform setup, service configuration, routing, subscriber management, QoS, and validation steps. Use these guides to test and verify RBFS Multiservice Edge solutions for PPPoE, IPoE, Redundancy (High Availability), and CGNAT deployments.

The following guides are available under the Reference Designs section:

- Multiservice Edge for IPoE Subscribers
- Multiservice Edge for PPPoE Subscribers
- Multiservice Edge with CGNAT for IPoE and PPPoE Subscribers
- Multiservice Edge with Redundancy for IPoE Subscribers

1. RBFS Multiservice Edge for PPPoE Subscribers

1.1. An Overview of RBFS Multiservice Edge for PPPoE Subscribers

This document provides information to validate the RBFS Multiservice Edge implementation using the PPPoE (Point-to-Point Protocol over Ethernet) network protocol. The guide contains information about general platform configuration, configuration of various access and routing protocols, subscriber management, Quality of Service (QoS) and troubleshooting. The document presents a single use case scenario and provides information specifically on how to validate this particular implementation and for more information on any specific application, refer to https://documents.rtbrick.com/.

This guide is not intended to be an exhaustive guide of all RBFS features and does not provide information on features such as Multicast, Lawful Intercept etc.

1.1.1. About RBFS Multiservice Edge

A Multiservice Edge is a BNG that contains all the BNG functionalities together in a single platform. Unlike the RBFS spine and leaf software images, where the functionalities are distributed to spine platforms and leaf platforms separately based on their roles, Multiservice Edge platform contains all functionalities together in a single platform.

The RtBrick Multiservice Edge is delivered as a Linux container that runs on platforms provided by the hardware ODM manufacturers. Platforms that support Multiservice Edge include Edgecore AGR400, and UfiSpace S9600. The RtBrick Multiservice Edge software runs on powerful bare-metal switches as an open BNG.

The BNG is designed to dynamically deliver the following services:

1. Discovering and managing subscriber sessions for PPPoE subscribers
2. Providing authentication, authorization and accounting (AAA)

The basic Multiservice Edge architecture for PPPoE subscribers is shown in Fig. 1.

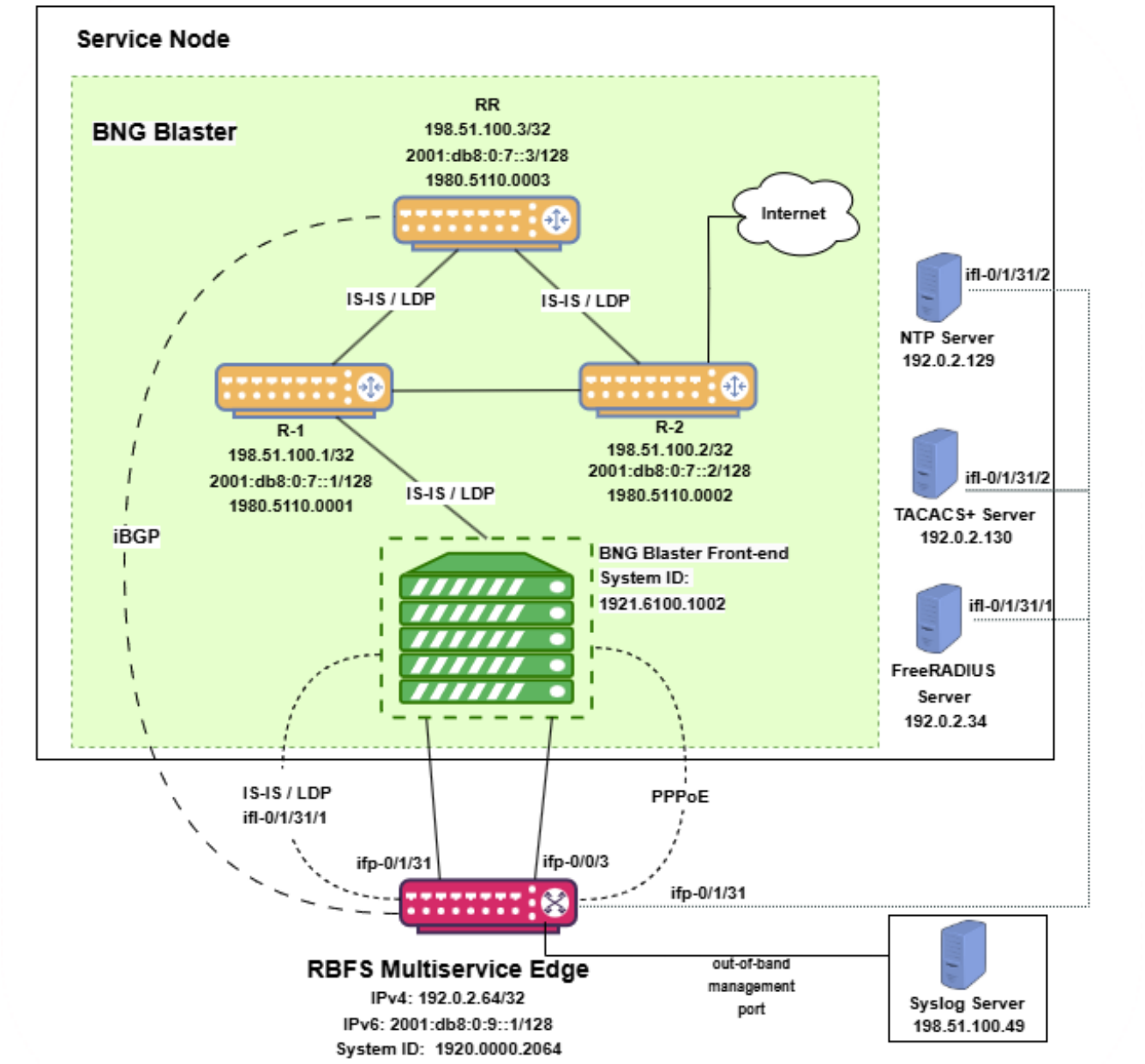


Figure 1. Topology setup with `Multiservice Edge` as a DUT

The figure above shows the topology setup with `Multiservice Edge` as a DUT (device under test) connected to BNG Blaster.

In this topology:

1. The Service Node is an Debian server on which a new container is spawned with the associated interfaces for running BNG Blaster tests.
2. BNG Blaster emulates both the routing and access functions. It is an open-source network tester developed by RtBrick to validate the performance and scalability of BNG devices. The BNG blaster can emulate both subscribers at scale and network elements such as routers. Therefore, it is possible to create complex topologies with just the DUT and a server that hosts the BNG Blaster, thereby minimizing the equipment needed to test the DU. For more information, refer to the BNG Blaster documentation. In this topology, BNG Blaster emulates PPPoE subscribers. The Service Node runs BNG Blaster. To use BNG Blaster, download and install the BNG Blaster

software onto the system that meets the tool's system requirements. The software can be downloaded from https://github.com/rtbrick/bngblaster/releases.

3. The topology emulates PPPoE subscribers and traffic between RBFS switch and the core network.
4. The objective of this topology is to demonstrate complete PPPoE subscriber emulating and service along with routing to connect to the network uplink.
5. R-1, R-2 and RR are simulated using BNG Blaster. The `Multiservice Edge` forms an IS-IS L1 adjacency with the BNG blaster to discover R-1, R-2, and RR. The RR is the route reflector that replays a full internet feed of IPv4 and IPv6 addresses, in total about 1.1 million prefixes. R-2 is the nexthop for all internet routes.
6. Through DCHP, an IPv4 address needs to be assigned to the out-of-band management port.

1.1.2. Deployment

A Multiservice Edge provides BNG functionality on a single bare-metal switch and eliminates the need to have a chassis-based system. It provides a low footprint and optimal power consumption based on BRCM chipsets, a compelling value proposition that has complete BNG and routing feature support.

Multiservice Edge runs on small form-factor temperature-hardened hardware that allows deployments in street cabinets.

The `rtbrick-toolkit` is a meta package that can be used to install all the tools needed to work with RBFS images (container or RtBrick Host installer) and the RBFS APIs.

For more information, see RBFS Installation Guide.

1.1.3. Using the RBFS CLI

Connect to the `Multiservice Edge` node.

```
$ ssh <multiservice-edge-management-ip> -l supervisor
supervisor@<multiservice-edge-management-ip>'s password:
```

The password for `multiservice-edge-management-ip` should be entered here.

As a result, the CLI prompt appears like this:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net:~ $
```

Open the RBFS CLI.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net:~ $ cli
```

The CLI has three different modes:

- `operation` mode is a read-only mode to inspect and analyze the system state
- `config` mode allows modifying the RBFS configuration
- `debug` mode provides advanced tools for troubleshooting

The `switch-mode` command allows switching between the different modes. The `show` commands allow inspecting the system state. The `set` and `delete` commands, which are only available in the configuration mode, allow modifying or deleting the current configuration. The `commit` command executes changes. RBFS provides a commit history that allows reviewing changes (`show commit log`) and restoring a previous configuration (`rollback`). There are also commands to `ping` destinations, `capture` network traffic, `save` the configuration to a file or `load` the configuration from a file.

The CLI supports abbreviating commands, provides suggestions by hitting the `[tab]` key and displays a context help by entering a `?`.

For more information on how to use the RBFS CLI, see RBFS CLI User Guide.

1.2. Configuration and Settings

1.2.1. Platform Configuration and Settings

This section provides information about the platform and how to set various required configurations for the platform.

Know your Device

The configurations provided in this reference design document (Multiservice Edge PPPoE implementation) are generated on the UfiSpace S9600-72XC platform. 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 [1GbE/10GbE/25GbE] and 8x100GE [40GbE/100GbE] high-speed ports with a switching capacity of up to 2.4Tbs.

The RBFS Multiservice Edge software is installed on top of the UfiSpace S9600-72XC.



Although the specific device used here is UfiSpace S9600-72XC, the configurations are exactly the same for any other device that supports the Multiservice Edge image.

For more information about the hardware specifications of UfiSpace S9600-72XC, see Hardware Specification.

Prerequisites

- Access to BNG Blaster, an open-source network testing platform for access and routing protocols. For information on obtaining and building BNG Blaster, see https://rtbrick.github.io/bngblaster/.
- Access to FreeRADIUS, a free RADIUS suite. For accessing FreeRADIUS, see https://freeradius.org/.

- Access to Syslog server.

Restore Configurations After Reboot

The system supports restoring the existing configurations after a reboot. This feature is turned off by default.

To enable restoring existing configurations, enter the `set system load-last-config true` command as shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> set system load-last-config
true
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> commit
```

For more information, see the section "Running Configuration" of the RBFS NOC Troubleshooting Guide.

1.2.2. General Configuration

To enable testing some basic primitives need to be configured. These general configurations include loopback interface for identifying and accessing the device on network, NTP for setting accurate time across a whole network of devices, TACACS+ for user authentication, user management for user configuration, license for accessing RBFS, Resmon for resource monitoring, and Syslog configurations for exporting the log message to the external log management server.

Configure License

Without any license installed on your system, you can evaluate RBFS for 7 days. You need to get an evaluation license or purchase an actual license within 7 days to use the full functionality of RBFS.

The following steps provide the commands to install an RBFS license key. For more information about license configuration, see Installing License.

Switch to config mode using the switch-mode config command to continue with the RBFS configurations.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> switch-mode config
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg>
```

Install the license encrypted string (that is received from RtBrick) using the RBFS CLI.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> set system license <license-
key>
```

RBFS license configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show config system license
AAAAWsfg&jdkfs4D34H5@2evf...
```

As shown below, the "show system license" command displays the expiration date for the current license.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show system license
License Validity:
  License index 1:
    Start date : Tue Feb 28 09:44:27 GMT +0000 2023
    End date   : Mon Mar 04 09:44:27 GMT +0000 2024
supervisor@rtbrick>multiservice-edge.rtbrick.net: op>
```

Configure Instance

Instance `default` is available by default without any configurations.

Create the instance `inband_mgmt` by entering the following commands.

```
set instance inband_mgmt
set instance inband_mgmt address-family ipv4 unicast
commit
```

The configurations of the instance `inband_mgmt` are shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config instance
inband_mgmt
{
  "rtbrick-config:instance": [
    {
      "name": "inband_mgmt",
      "address-family": [
        {
          "afi": "ipv4",
          "safi": "unicast"
        }
      ]
    }
  ]
}
```

Below are the configurations available for the available instances.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show instance detail
Instance: default
  Instance ID: 0
```

```
  State: Active
  AFI        SAFI             State
  ipv4       unicast          Active
  ipv4       multicast        Active
  ipv4       labeled-unicast  Active
  ipv6       unicast          Active
  ipv6       multicast        Active
  ipv6       labeled-unicast  Active
  mpls       unicast          Active
Instance: inband_mgmt
  Instance ID: 3
  State: Active
  AFI        SAFI             State
  ipv4       unicast          Active
```

Configure Loopback Interface

Loopback Interface configuration is required as it is the best way to identify a network device and is always reachable. Also, protocols use the loopback address to determine protocol-specific properties for the device.

The following steps provide the commands to configure the loopback interface. For more information about Loopback Interface configuration, see Interfaces User Guide.

Configure loopback interface on the device.

```
set interface lo-0/0/1 unit 0 address ipv4 192.0.2.64/32
set interface lo-0/0/1 unit 1 address ipv4 192.0.2.74/32
set interface lo-0/0/1 unit 2 instance inband_mgmt
set interface lo-0/0/1 unit 2 address ipv4 192.0.2.128/32
set interface lo-0/0/1 unit 3 instance inband_mgmt
set interface lo-0/0/1 unit 3 address ipv4 192.0.2.131/32
commit
```



The configuration commands should be followed by the `commit` command to save the configurations into the device.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> commit
```

Loopback Interface configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config interface lo-0/0/1
{
  "rtbrick-config:interface": [
    {
      "name": "lo-0/0/1",
      "unit": [
```

```
        {
          "unit-id": 0,
          "address": {
            "ipv4": [
              {
                "prefix4": "192.0.2.64/32"
              }
            ]
          }
        },
        {
          "unit-id": 1,
          "address": {
            "ipv4": [
              {
                "prefix4": "192.0.2.74/32"
              }
            ]
          }
        },
        {
          "unit-id": 2,
          "instance": "inband_mgmt",
          "address": {
            "ipv4": [
              {
                "prefix4": "192.0.2.128/32"
              }
            ]
          }
        },
        {
          "unit-id": 3,
          "instance": "inband_mgmt",
          "address": {
            "ipv4": [
              {
                "prefix4": "192.0.2.131/32"
              }
            ]
          }
        }
      ]
    }
  ]
}
```

Configure IP Addresses for Core Interfaces

Enter the following commands to configure IP addresses for the core interfaces.

```
set interface ifp-0/1/31 unit 10
set interface ifp-0/1/31 unit 10 vlan 10
set interface ifp-0/1/31 unit 10 address ipv4 192.0.2.1/27
set interface ifp-0/1/31 unit 10 address ipv6 2001:db8::1/64
set interface ifp-0/1/31 unit 100
set interface ifp-0/1/31 unit 100 vlan 100
set interface ifp-0/1/31 unit 100 address ipv4 192.0.2.33/27
set interface ifp-0/1/31 unit 200
set interface ifp-0/1/31 unit 200 instance inband_mgmt
set interface ifp-0/1/31 unit 200 vlan 200
set interface ifp-0/1/31 unit 200 address ipv4 192.0.2.97/27
commit
```

Below configuration shows the IP address configurations for the core interfaces.

```
{
    "rtbrick-config:interface": [
      {
        "name": "ifp-0/1/31",
        "unit": [
          {
            "unit-id": 10,
            "vlan": 10,
            "address": {
              "ipv4": [
                {
                  "prefix4": "192.0.2.1/27"
                }
              ],
              "ipv6": [
                {
                  "prefix6": "2001:db8::1/64"
                }
              ]
            }
          },
          {
            "unit-id": 100,
            "vlan": 100,
            "address": {
              "ipv4": [
                {
                  "prefix4": "192.0.2.33/27"
                }
              ]
            }
          },
          {
            "unit-id": 200,
```

```
            "instance": "inband_mgmt",
            "vlan": 200,
            "address": {
              "ipv4": [
                {
                  "prefix4": "192.0.2.97/27"
                }
              ]
            }
          }
        ]
      }
    ]
  }
}
```

Configure Static Routes to Enable Reachability to the NTP and TACACS Servers

Below are static routes configured to enable reachability to the NTP (192.0.2.129) and TACACS (192.0.2.130) servers. On the Service Node, 192.0.2.98 is the interface address on VLAN 200. It is explained later in this document how to configure IP addresses on Service Node. For details, see Configuring Interfaces on the Service Node for NTP and TACACS Connectivity from Multiservice Edge.

```
set instance inband_mgmt static route ipv4 192.0.2.129/32 unicast np1
set instance inband_mgmt static route ipv4 192.0.2.130/32 unicast np1
set instance inband_mgmt static nexthop-profile np1
set instance inband_mgmt static nexthop-profile np1 nexthop 192.0.2.98
set instance inband_mgmt static nexthop-profile np1 lookup-instance inband_mgmt
set instance inband_mgmt static nexthop-profile np1 lookup-afi ipv4
set instance inband_mgmt static nexthop-profile np1 lookup-safi unicast
commit
```

The configuration of the static routes is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config instance
inband_mgmt static
{
  "rtbrick-config:static": {
    "route": {
      "ipv4": [
        {
          "prefix4": "192.0.2.129/32",
          "safi": "unicast",
          "nexthop-profile": "np1"
        },
        {
          "prefix4": "192.0.2.130/32",
```

```
          "safi": "unicast",
          "nexthop-profile": "np1"
        }
      ]
    },
    "nexthop-profile": [
      {
        "name": "np1",
        "nexthop": "192.0.2.98",
        "lookup-instance": "inband_mgmt",
        "lookup-afi": "ipv4",
        "lookup-safi": "unicast"
      }
    ]
  }
}
```

Configure NTP

Configuring NTP (Network Time Protocol) provides time synchronization across a whole network of devices. An NTP network consists of devices (clients) that are to be synchronized with the NTP server that provides accurate time to the client devices.

The following steps provide the commands to configure Network Time Protocol (NTP) for the device. For more information about NTP configuration, see NTP User Guide.

Enabling NTP Service:

To access the NTP service running in the RtBrick Host, this service has to be enabled in inband-management. On configuring this, the hosts reachable in inband instance via the physical interface can access this service.

Configure NTP server and NTP service on the device.

```
set system ntp server ntp1
set system ntp server ntp1 ipv4-address 192.0.2.129
set inband-management instance inband_mgmt
set inband-management instance inband_mgmt ntp true
commit
```

NTP configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config inband-management
{
    "rtbrick-config:inband-management": {
      "instance": [
        {
          "name": "inband_mgmt",
          "ntp": "true"
```

```
        }
      ]
    }
  }
```

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config system ntp
{
    "rtbrick-config:ntp": {
      "server": [
        {
          "server-name": "ntp1",
          "ipv4-address": "192.0.2.129"
        }
      ]
    }
  }
```

User Authentication

RBFS supports user authentication through a centralized TACACS+ server and with a local authentication system. The following authentication process typically occurs when a user attempts to access the network.

1. When a user logs in through SSH, the SSH Daemon (sshd) invokes the Pluggable Authentication Module (PAM) to trigger the authentication process.
2. PAM requests TACACS+ authentication (except for the user with the supervisor privileges).
3. TACACS+ server provides 'grant access' node if the user authentication is successful.
4. If the user is not allowed using the TACACS+ authentication, it is required to undergo an additional authentication phase. PAM looks up local users. Upon successful authentication, PAM generates RTB PAM token; includes user role in 'scope'.

Define Users on TACACS+ Server

Administrators need to define users and associate them with the predefined roles on the TACACS+ server. Optionally, RBFS CLI commands can be restricted using the rtb-allow-cmds and rtb-deny-cmds.

The tac_plus.conf file contains configuration information for the tac_plus(tacacs+) daemon. This file is stored at the following location:

/etc/tacacs+/tac_plus.conf

To view the TACACS+ server configuration file, enter the following command.

```
sudo cat /etc/tacacs+/tac_plus.conf
```

For more information about TACACS+ server configuration, see

https://manpages.ubuntu.com/manpages/trusty/man5/tac_plus.conf.5.html

This Reference Design document uses the default local user `supervisor` for the configurations, whereas other users, defined in the TACACS server, can log into RBFS by using their usernames and passwords.

The following TACACS+ configuration shows the details of the TACACS users.

➔ Click here to download the `tac_plus.conf` file.

Configure TACACS+ on RBFS

After defining the users on the TACACS+ server, configure the TACACS+ server on Multiservice Edge. This configuration allows the remote TACACS+ server to communicate with the Multiservice Edge and validate user access on the network.

The following steps provide the commands to configure TACACS+. For more information about TACACS+ configuration, see Configure TACACS+ on RBFS.

To access the TACACS+ service running in the RtBrick Host, this service has to be enabled in inband management. On configuring this, the hosts reachable in inband instance through the physical interface can access this service.

```
set system secure-management-status true
set system authorization tacacs 192.0.2.130 inband secret-plain-text
RtBrick_Little_Secret
set inband-management instance inband_mgmt tacacs true
commit
```

In the above configuration, the command `set inband-management instance inband_mgmt tacacs true` is used to enable TACACS+ under the instance called `inband_mgmt`.

TACACS+ configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config system
authorization
{
  "rtbrick-config:authorization": {
    "tacacs": [
      {
        "ipv4-address": "192.0.2.130",
        "type": "inband",
        "secret-encrypted-text":
"$22464b2c7336cfe71e596c447be28d598b9b7b37f92faea157fd5058e5fe0d769"
      }
    ]
  }
}
```

Configuration for enabling TACACS+ under the instance `inband_mgmt` is shown below:

```
"rtbrick-config:inband-management": {
      "instance": [
        {
          "name": "inband_mgmt",
          "tacacs": "true"
        }
      ]
    },
```

Enabling TACACS+ Service on the Service Node

Enter the following commands to enable the TACACS service on the Service Node.

```
~$ sudo /bin/systemctl enable tacacs_plus.service
~$ sudo /bin/systemctl start tacacs_plus.service
```

Validating TACACS+ authentication

The following scenario shows a successful authentication for the user `bob` with password `bob`.

```
~$ ssh bob@multiservice-edge.rtbrick.net
bob@multiservice-edge.rtbrick.net's password:
Last login: Mon Apr 3 16:13:40 2023 from multiservice-edge.rtbrick.net
bob@rtbrick>multiservice-edge.rtbrick.net: op>
```

The following scenario shows an unsuccessful password authentication for the user `bob` with password `bob123`.

```
~$ ssh bob@rtbrick>multiservice-edge.rtbrick.net:
bob@multiservice-edge.rtbrick.net's password:
Permission denied, please try again.
bob@multiservice-edge.rtbrick.net's password:
```

The following scenario shows an unsuccessful authentication for an undefined user `frank`.

```
~$ ssh frank@rtbrick>multiservice-edge.rtbrick.net:
frank@multiservice-edge.rtbrick.net's password:
Permission denied, please try again.
frank@multiservice-edge.rtbrick.net's password:
accounting file = /var/log/tac_plus.acct
key = RtBrick_Little_Secret
```

Configure User Management

Configuring Local User Management enables administrators to create, manage, and secure the users and groups. It allows creation of privileges that are configurable for user-defined and predefined roles.

The following steps provide the commands to configure user management. For more information about license configuration, see Local User Management.

1. To create a role, configure the RBAC privilege and the command privilege. To configure the RBAC privilege for both table and object:

```
set system user admin role supervisor
set system user admin shell /bin/bash
set system user admin password-hashed-text
$6$XNkmuMRI.5.R/NBJ$XDfZec7gEM3z/3lYn8mDDWimRZ/68xawia.pTMdrGqoYHEE3nWHB08DeaPNQTwHW6W
jB1aX6.xjYjh8CNCy4g1
commit
```

For information about Configuring hashed password, see Configure Hashed Password.

Authentication configuration of a password hashed text and an SSH public key is shown below:

```
{
  "ietf-restconf:data": {
    "rtbrick-config:system": {
      "user": [
        {
          "username": "admin",
          "shell": "/usr/local/bin/cli",
          "password-hashed-text":
"$5$L2DaOYYuddhBV$9RA5MX9RQzLC9fIKJzbnoFBb88w9rkSXl7GVrVJ9PY7",
          "ssh-pub-key": [
            "ssh-rsa AAAAWsfg&jdkfs4D34H5@2evf....."
            ]
        }
      ]
    }
  }
}
```

Configure Syslog

RBFS supports sending log messages to a Syslog server. The Syslog configuration can be performed in RBFS.

To configure logging for `bgp` by using Syslog, enter the following commands.

```
set log module bgp
set log module bgp level debug
set log module bgp plugin-alias
set log module bgp plugin-alias alias-name syslog
set log module bgp plugin-alias level debug
commit
```



For event logging, RBFS supports Graylog and Syslog. Graylog must be disabled in order to enable Syslog. In addition, Graylog attributes must be replaced with Syslog attributes.

Sylog configuration for the module `bgp` is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config log
{
  "rtbrick-config:log": {
    "module": [
      {
        "module-name": "bgp",
        "level": "debug",
        "plugin-alias": {
          "alias-name": "syslog",
          "level": "debug"
        }
      }
    ]
  }
}
```

Accessing the RtBrick Host to Configure Syslog

The steps described in this section are performed on the RtBrick Host. For logging into the RtBrick Host, use SSH port 1022.

```
ssh supervisor@<multiservice-edge-management-ip> -p 1022
```

After logging into the RtBrick Host, go to the following location of CtrlD and edit the `config.json` file.

- `/etc/rtbrick/ctrld/config.json`

Specify the Syslog configurations as shown below in the `config.json` file.

```
{
    "rbms_enable": false,
    "graylog_enable": false,
    "syslog_enable": true,
```

```
    "syslog_network": "udp",
    "syslog_urls": [
      "198.51.100.49:516"
    ],
    "syslog_severity_level": 7,
    "auth_enabled": false
}
```



- For documentation purposes, the IP address `198.51.100.49` has been used as the IP address of the Syslog endpoint. This IP address should be updated with the actual Syslog server's IP address.
- Syslog messages can be transported using UDP or TCP protocol. In this configuration, Syslog messages are transported using `udp`.

After making configuration changes in the `config.json`, restart CtrlD service as shown below.

```
supervisor@onl>multiservice-edge.rtbrick.net:~ $ sudo service rtbrick-ctrld restart
[sudo] password for supervisor:
[ ok ] Stopping rtbrick ctrld service:.
[ ok ] Starting rtbrick ctrld service:.
```

Monitor Resources (Resmon)

Resource monitoring enables administrators to collect and analyze the health information and usage data of various hardware resources such as CPU, memory, processes, disks, sensors, optics, and so on.

Run `show cpu usage`, `show memory usage` and `show disk usage` to see the CPU, memory and disk utilization respectively.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show cpu usage
Name         Total       User     System       Nice   I/O Wait       Idle        IRQ   Soft IRQ
cpu            11%         9%         1%         0%         0%        88%         0%         0%
cpu0           10%         8%         1%         0%         0%        89%         0%         0%
cpu1           44%        43%         1%         0%         0%        55%         0%         0%
cpu2           34%        32%         2%         0%         0%        66%         0%         0%
cpu3           11%         9%         2%         0%         0%        89%         0%         0%
cpu4            3%         1%         2%         0%         0%        96%         0%         0%
cpu5            4%         3%         1%         0%         0%        96%         0%         0%
cpu6           13%        11%         2%         0%         0%        87%         0%         0%
cpu7           24%        22%         1%         0%         0%        75%         0%         0%
cpu8            4%         2%         2%         0%         0%        95%         0%         0%
cpu9            2%         1%         1%         0%         0%        97%         0%         0%
cpu10           6%         4%         2%         0%         0%        93%         0%         0%
cpu11           3%         2%         1%         0%         0%        96%         0%         0%
cpu12           8%         6%         2%         0%         0%        91%         0%         0%
cpu13           3%         3%         0%         0%         0%        96%         0%         0%
cpu14           6%         4%         2%         0%         0%        93%         0%         0%
cpu15           8%         5%         2%         0%         0%        91%         0%         0%
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show memory usage
Name     Total           Used            Free         Shared       Buffers      Cached
```

```
RAM      31.03 GiB        8.51 GiB         17.04 GiB    1.19 GiB     112.66 MiB   5.37 GiB
SWAP     0 bytes          0 bytes          0 bytes      n/a          n/a          n/a
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show disk usage
Filesystem               Type             Size         Used         Available    Mountpoint          Usage %
none                     tmpfs            492 KiB      0 bytes      492 KiB      /dev                0.0
tmpfs                    tmpfs            15.51 GiB    17.23 MiB    15.5 GiB     /run                0.11
tmpfs                    tmpfs            6 GiB        828.75 MiB   5.19 GiB     /shm                13.49
tmpfs                    tmpfs            15.51 GiB    182.96 MiB   15.33 GiB    /dev/shm            1.15
tmpfs                    tmpfs            5 MiB        0 bytes      5 MiB        /run/lock           0.0
devtmpfs                 devtmpfs         1 MiB        0 bytes      1 MiB        /dev/mem            0.0
/dev/sda10               ext4             15.62 GiB    50.61 MiB    14.76 GiB    /var/log            0.33
/dev/sda6                ext4             29.4 GiB     4.24 GiB     23.65 GiB    /platform           15.2
tmpfs                    tmpfs            3.1 GiB      0 bytes      3.1 GiB      /run/user/1000      0.0
tmpfs                    tmpfs            3.1 GiB      0 bytes      3.1 GiB      /run/user/1001      0.0
tmpfs                    tmpfs            15.51 GiB    0 bytes      15.51 GiB    /sys/fs/cgroup      0.0
/dev/sda11               ext4             43.79 GiB    51.89 MiB    41.49 GiB    /var/crash          0.12
tmpfs                    tmpfs            3.1 GiB      1.02 MiB     3.1 GiB      /var/run-ext/onl/r  0.03
/var/cache/rtbrick/imag  overlay          29.4 GiB     4.24 GiB     23.65 GiB    /                   15.2
```

The show command can also be used to view other resource details. For information about the resmon configuration and operational commands, see the RBFS Resource Monitoring Guide.

1.3. Protocol Configurations

This validated solution design topology uses IS-IS as the interior gateway protocol to distribute IP routing information among the routers in an Autonomous System (AS). The Label Distribution Protocol (LDP) is used to exchange label mapping information for MPLS traffic. And, iBGP is used for exchanging routing and reachability information within ASs.

One thus needs to configure the following protocols:

- IS-IS: To ensure IP connectivity on the core network.
- LDP: To establish MPLS LSP tunnels for MPLS data transmission on the network.
- iBGP: To exchange routing information within an AS.

1.3.1. Configure IS-IS

The following steps provide the commands to execute various IS-IS protocol functionalities. For more detailed information about IS-IS configuration, see IS-IS User Guide.

1. Configure IS-IS system-id, area, hostname and interfaces.

```
set instance default protocol isis system-id 1920.0000.2064
set instance default protocol isis area 49.0001/24
set instance default protocol isis hostname multiservice-edge
set instance default protocol isis interface ifl-0/1/31/10
set instance default protocol isis interface ifl-0/1/31/10 type point-to-point
set instance default protocol isis interface ifl-0/1/31/10 level-2 adjacency-disable
true
set instance default protocol isis interface lo-0/0/1/0
set instance default protocol isis interface lo-0/0/1/0 passive true
```

```
commit
```

IS-IS instance configuration on interface is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show config instance default
protocol isis
{
  "rtbrick-config:isis": {
    "system-id": "1920.0000.2064",
    "area": [
      "49.0001/24"
      ],
    "hostname": "multiservice-edge",
    "interface": [
      {
        "name": "ifl-0/1/31/10",
        "type": "point-to-point",
        "level-2": {
          "adjacency-disable": "true"
        }
      },
      {
        "name": "lo-0/0/1/0",
        "passive": "true"
      }
    ]
  }
}
```

1.3.2. Configure LDP on the Interfaces

The following steps provide the commands to execute various LDP functionalities. For more detailed information about LDP configuration, see *LDP User Guide*.

1. Configure LDP on the router interface.

```
set instance default protocol ldp router-id 192.0.2.64
set instance default protocol ldp interface ifl-0/1/31/10
set instance default protocol ldp interface lo-0/0/1/0
commit
```

Configuration for LDP on the interface is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config instance default
protocol ldp
{
  "rtbrick-config:ldp": {
```

```
    "router-id": "192.0.2.64",
    "interface": [
      {
        "name": "ifl-0/1/31/10"
      },
      {
        "name": "lo-0/0/1/0"
      }
    ]
  }
}
```

1.3.3. Configure BGP

The following steps provide the commands to execute the various BGP functionalities quickly. For more detailed information about BGP configuration, see BGP User Guide.

1. Configure BGP local AS, router-id, and hostname

```
set instance default protocol bgp local-as 4200000001
set instance default protocol bgp router-id 192.0.2.64
set instance default protocol bgp hostname multiservice-edge
commit
```

BGP local AS, router-id, and hostname configurations are shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config instance default
protocol bgp
{
  "rtbrick-config:bgp": {
    "local-as": 4200000001,
    "hostname": "multiservice-edge",
    "router-id": "192.0.2.64",

<....>
```

2. Enable the IPv4 and IPv6 address families which are to be supported on the specific BGP instance.

```
set instance default protocol bgp address-family ipv4 unicast
set instance default protocol bgp address-family ipv4 unicast resolve-nexthop safi
labeled-unicast
set instance default protocol bgp address-family ipv6 labeled-unicast
set instance default protocol bgp address-family ipv6 unicast
set instance default protocol bgp address-family ipv6 unicast resolve-nexthop safi
labeled-unicast
```

```
commit
```

BGP address family configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config instance default
protocol bgp address-family
{
  "rtbrick-config:address-family": [
    {
      "afi": "ipv4",
      "safi": "unicast",
      "resolve-nexthop": {
        "safi": "labeled-unicast"
      }
    },
    {
      "afi": "ipv6",
      "safi": "labeled-unicast"
    },
    {
      "afi": "ipv6",
      "safi": "unicast",
      "resolve-nexthop": {
        "safi": "labeled-unicast"
      }
    }
  ]
}
```

3. Create the peer group with the specific remote AS configurations and the address family that is to be negotiated with the peer which will be attached to the peer group later.

```
set instance default protocol bgp peer-group RR
set instance default protocol bgp peer-group RR remote-as 4200000001
set instance default protocol bgp peer-group RR address-family ipv4 unicast
set instance default protocol bgp peer-group RR address-family ipv6 labeled-unicast
set instance default protocol bgp peer-group RR address-family ipv6 unicast
commit
```

BGP peer-group configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config instance default
protocol bgp peer-group
{
  "rtbrick-config:peer-group": [
    {
      "pg-name": "RR",
      "remote-as": 4200000001,
      "address-family": [
        {
          "afi": "ipv4",
          "safi": "unicast"
        },
        {
          "afi": "ipv6",
          "safi": "labeled-unicast"
        },
        {
          "afi": "ipv6",
          "safi": "unicast"        }
      ]
    }
  ]
}
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg>
```

4. Add a BGP peer and associate it with the specific peer group.

```
set instance default protocol bgp peer
set instance default protocol bgp peer ipv4 198.51.100.3 192.0.2.64
set instance default protocol bgp peer ipv4 198.51.100.3 192.0.2.64 peer-group RR
commit
```

Configuration for adding a BGP peer and associating it with a peer group is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config instance default
protocol bgp peer
{
  "rtbrick-config:peer": {
    "ipv4": [
      {
        "peer-address": "198.51.100.3",
        "update-source": "192.0.2.64",
        "peer-group": "RR"
      }
    ]
  }
```

```
}
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg>
```

5. Configure the IPv6 unicast address family with `send-label` as true, then address-family IPv6 labeled-unicast gets negotiated with the peer.

```
set instance default protocol bgp peer-group RR address-family ipv6 unicast send-label
true
commit
```

The following configuration shows the BGP IPv6 unicast address family with `send-label` as true.

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

6. Set the resolve-nexthop, if the BGP nexthop attribute of the BGP routes needs to be resolved under ipv4/ipv6 labeled-unicast routing table. It configures only resolve-nexthop safi. Based on the nexthop-type (ipv4 or ipv6), it gets looked up into either IPv4 labeled-unicast or IPv6 labeled-unicast.

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

Resolve nexthop configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config instance default
protocol bgp address-family ipv4 unicast resolve-nexthop
{
  "rtbrick-config:resolve-nexthop": {
    "safi": "labeled-unicast"
  }
}
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg>
```

7. To redistribute the routes (belonging to a specific source) into BGP, execute the following command. The following command redistributes `direct` routes into BGP.

```
set instance default protocol bgp address-family ipv4 unicast redistribute direct
commit
```

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config instance default
protocol bgp address-family ipv4 unicast redistribute
{
  "rtbrick-config:redistribute": [
    {
      "source": "direct"
    }
  ]
}
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg>
```

1.3.4. Configuring the Service Node

Configuring Interfaces on the Service Node for RADIUS Connectivity from Multiservice Edge

To configure interfaces on the Service Node for RADIUS connectivity from Multiservice Edge, enter the following commands:

```
sudo ip link add link SN1-3-C1 name SN1-3-C1.100 type vlan id 100
sudo ifconfig SN1-3-C1.100 192.0.2.34/27
```



`SN1-3-C1` is the internal nomenclature that denotes the interface name on Service Node that connects to the Multiservice Edge..

Configuring Interfaces on the Service Node for NTP and TACACS Connectivity from Multiservice Edge

To configure interfaces on the Service Node for NTP and TACACS connectivity from Multiservice Edge, enter the following commands:

```
sudo ip link add link SN1-3-C1 name SN1-3-C1.200 type vlan id 200
sudo ifconfig SN1-3-C1.200 192.0.2.98/27
sudo ifconfig lo:1 192.0.2.129 netmask 255.255.255.255 up
sudo ifconfig lo:2 192.0.2.130 netmask 255.255.255.255 up
```

Configuring Routes on the Service Node

To configure routes on the Service Node that provides reachability to the RADIUS, TACACS and NTP servers, enter the following commands:

```
sudo ip route add 192.0.2.74/32 via 192.0.2.33
sudo ip route add 192.0.2.131/32 via 192.0.2.97
sudo ip route add 192.0.2.128/32 via 192.0.2.97
```

BNG Blaster Configuration for Protocols

BNG Blaster is an open-source network testing platform for access and routing protocols. It can emulate massive PPPoE and IPoE (DHCP) subscribers including IPTV, and L2TP (LNS). There are various routing protocols supported such as IS-IS and BGP. So, one can use this platform for end-to-end BNG and non-BNG router testing.

For more information about BNG Blaster, see https://github.com/rtbrick/bngblaster

For information about installing BNG Blaster, see https://rtbrick.github.io/bngblaster/install.html

Downloading the Blaster Configuration File

The following is the configuration file that is used in BNG Blaster for validating PPPoE, BGP, IS-IS, and LDP.

➔ Click here to download the BNG Blaster configuration file (`blaster.json`).

Generating Supporting Files for Protocols

- **Generating BGP Internet Prefixes**

Enter the following commands to generate BGP internet prefixes.

```
bgpupdate -f internet.bgp -a 4200000001 -n 198.51.100.2 -N 1 -p 172.16.0.0/24 -P
1000000
bgpupdate -f internet.bgp -a 4200000001 -n 198.51.100.2 -N 1 -p 2004::/48 -m 10000 -M
5 -P 150000 --append
```

Ensure that the command execution has finished (as shown below) before continuing.

```
[2023-04-05 10:19:32][INFO   ] init 1000000 IPv4 prefixes
[2023-04-05 10:19:56][INFO   ] open file internet.bgp (replace)
[2023-04-05 10:25:31][INFO   ] finished
```

After the generation of the `internet.bgp` file, the "raw-update-file" attribute of the `blaster.json` file needs to be updated as follows:

```
"raw-update-file": "/home/supervisor/internet.bgp"
```



For more information about downloading the blaster.json file, see section Downloading the Blaster Configuration File.

- **Generating MRT FIle for IS-IS**

Below is the JSON file (`isis_3node.json`) which is used to simulate R-1, R-2, and RR on BNG Blaster.

➔ Click here to download the `isis_3node.json` file.

This JSON file needs to be converted to MRT format using the following command:

```
lspgen -r isis_3node.json -m isis.mrt
```

After converting the file to `isis.mrt`, it needs to be updated in the IS-IS section in the `blaster.json` file as shown below.

```
"mrt-file": "/home/supervisor/isis.mrt"
```

- Generate labels for the IS-IS prefixes using "ldpupdate" command as shown below:

```
ldpupdate -l 192.0.2.65 -p 198.51.100.1/32 -P 3 -M 3 -f out.ldp
```

The detail of the generated file needs to be added to the LDP section in the `blaster.json` file as shown below:

```
"raw-update-file": "/home/supervisor/out.ldp"
```

Starting BNG Blaster

In the following command line string, a BNG Blaster instance is started and the `blaster.json` file is used.

```
sudo bngblaster -C blaster.json -I
```

The **-C blaster.json** argument specifies the blaster configuration file. The -I flag enables the interactive blaster UI.

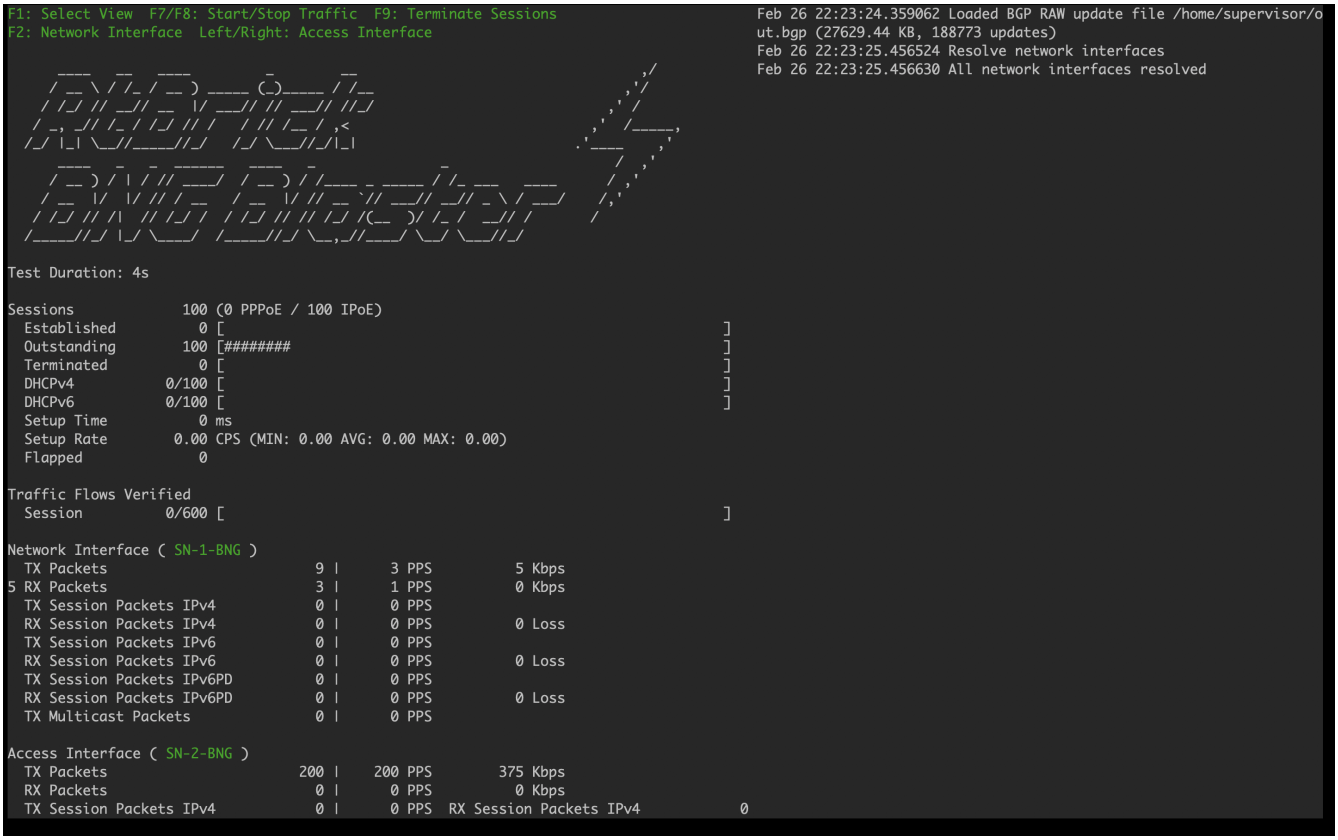


Figure 2. Starting BNG Blaster

1.3.5. Validating Protocols on RBFS Multiservice Edge

Validating IS-IS Adjacency, Routes and Reachability

Run the following command to show IS-IS adjacency.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show isis neighbor
Instance: default
  Interface           System            Level   State   Type    Up since               Expires
  ifl-0/1/31/10       BNG-Blaster       L1      Up      P2P     Wed Mar 29 05:23:24    in 22s 242106us
```

After configuring IS-IS protocol, check the IPv4 unicast routes, populated by IS-IS using the following command:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show route ipv4 unicast source isis instance default
Instance: default, AFI: ipv4, SAFI: unicast
Prefix/Label                          Source          Pref    Next Hop                          Interface

192.0.2.2/27                          isis            15      192.0.2.2                         ifl-0/1/31/10
192.0.2.65/32                         isis            15      192.0.2.2                         ifl-0/1/31/10
198.51.100.1/32                       isis            15      192.0.2.2                         ifl-0/1/31/10
198.51.100.2/32                       isis            15      192.0.2.2                         ifl-0/1/31/10
198.51.100.3/32                       isis            15      192.0.2.2                         ifl-0/1/31/10
198.51.100.101/31                     isis            15      192.0.2.2                         ifl-0/1/31/10
198.51.100.103/31                     isis            15      192.0.2.2                         ifl-0/1/31/10
198.51.100.105/31                     isis            15      192.0.2.2                         ifl-0/1/31/10
```

Ping the address 192.0.2.65 as follows:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> ping 192.0.2.65
68 bytes from 192.0.2.65: icmp_seq=1 ttl=64 time=9.0436 ms
68 bytes from 192.0.2.65: icmp_seq=2 ttl=64 time=2.0959 ms
68 bytes from 192.0.2.65: icmp_seq=3 ttl=64 time=4.7229 ms
68 bytes from 192.0.2.65: icmp_seq=4 ttl=64 time=9.2496 ms
68 bytes from 192.0.2.65: icmp_seq=5 ttl=64 time=2.6149 ms
Statistics: 5 sent, 5 received, 0% packet loss
```

Validating LDP Adjacency, Routes and Reachability

Run the following commands to show LDP neighbor and LDP session.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show ldp neighbor
Instance: default
  Interface           LDP ID              Transport IP  Up Since                  Expires
  ifl-0/1/31/10        192.0.2.65:0        192.0.2.65    Wed Mar 29 05:21:11       in 11s
```

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show ldp session
Instance: default
  LDP ID            Peer IP           State             Up/Down           FECRcvd   FECSent
  192.0.2.65:0      192.0.2.65        Operational       0d:03h:55m:49s          5         5
```

After configuring the LDP protocol, check the IPv4 labeled unicast routes, populated by LDP using the following command:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op>  show route ipv4 labeled-unicast source ldp
Instance: default, AFI: ipv4, SAFI: labeled-unicast
Prefix/Label                            Source           Pref    Next Hop                                Interface
Label
192.0.2.2/27                            ldp              9       192.0.2.2                               ifl-0/1/31/10
-
192.0.2.65/32                           ldp              9       192.0.2.2                               ifl-0/1/31/10
-
198.51.100.1/32                         ldp              9       192.0.2.2                               ifl-0/1/31/10
10000
198.51.100.2/32                         ldp              9       192.0.2.2                               ifl-0/1/31/10
10001
198.51.100.3/32                         ldp              9       192.0.2.2                               ifl-0/1/31/10
10002
```

Ping the labeled unicast address 198.51.100.1 as follows:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> ping 198.51.100.1 instance
default afi ipv4 safi labeled-unicast
68 bytes from 198.51.100.1: icmp_seq=1 ttl=64 time=6.3289 ms
68 bytes from 198.51.100.1: icmp_seq=2 ttl=64 time=2.8249 ms
68 bytes from 198.51.100.1: icmp_seq=3 ttl=64 time=1.8587 ms
68 bytes from 198.51.100.1: icmp_seq=4 ttl=64 time=5.9599 ms
68 bytes from 198.51.100.1: icmp_seq=5 ttl=64 time=4.3811 ms
```

```
Statistics: 5 sent, 5 received, 0% packet loss
```



The command argument `labeled-unicast` takes the ICMP requests through a labeled path while validating IP connectivity and hence, it prepends an MPLS label.

Validating BGP Adjacency, Routes and Reachability

Run the following commands to show BGP session and state.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show bgp peer
Instance: default
  Peer                                  Remote AS    State         Up/Down Time              PfxRcvd
PfxSent
  198.51.100.3                          4200000001   Established   0d:00h:01m:24s            1150000             5
```

After configuring BGP, check the IPv4 unicast routes, populated by BGP using the following command:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show route ipv4 unicast source bgp instance default
Instance: default, AFI: ipv4, SAFI: unicast
Prefix/Label                          Source           Pref    Next Hop                              Interface
172.16.0.0/24                         bgp              200     198.51.100.2                          ifl-0/1/31/10
172.16.1.0/24                         bgp              200     198.51.100.2                          ifl-0/1/31/10
172.16.2.0/24                         bgp              200     198.51.100.2                          ifl-0/1/31/10
172.16.3.0/24                         bgp              200     198.51.100.2                          ifl-0/1/31/10
172.16.4.0/24                         bgp              200     198.51.100.2                          ifl-0/1/31/10
172.16.5.0/24                         bgp              200     198.51.100.2                          ifl-0/1/31/10
172.16.6.0/24                         bgp              200     198.51.100.2                          ifl-0/1/31/10
172.16.7.0/24                         bgp              200     198.51.100.2                          ifl-0/1/31/10
172.16.8.0/24                         bgp              200     198.51.100.2                          ifl-0/1/31/10
172.16.9.0/24                         bgp              200     198.51.100.2                          ifl-0/1/31/10

<...>
```

This command will list all the 1 million IPv4 BGP internet prefixes.

Pinging an IPv4 route (source: bgp) from the multiservice-edge.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> ping 172.16.1.0
68 bytes from 172.16.1.0: icmp_seq=1 ttl=64 time=6.0527 ms
68 bytes from 172.16.1.0: icmp_seq=2 ttl=64 time=6.2893 ms
68 bytes from 172.16.1.0: icmp_seq=3 ttl=64 time=2.5573 ms
68 bytes from 172.16.1.0: icmp_seq=4 ttl=64 time=4.6964 ms
68 bytes from 172.16.1.0: icmp_seq=5 ttl=64 time=5.6455 ms
Statistics: 5 sent, 5 received, 0% packet loss
```

Check the IPv6 unicast routes, populated by BGP using the following command:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show route ipv6 unicast source bgp instance default
Instance: default, AFI: ipv6, SAFI: unicast
```

```
Prefix/Label                    Source          Pref    Next Hop                    Interface
2004::/48                       bgp             200     198.51.100.2                ifl-0/1/31/10
2004:0:1::/48                   bgp             200     198.51.100.2                ifl-0/1/31/10
2004:0:2::/48                   bgp             200     198.51.100.2                ifl-0/1/31/10
2004:0:3::/48                   bgp             200     198.51.100.2                ifl-0/1/31/10
2004:0:4::/48                   bgp             200     198.51.100.2                ifl-0/1/31/10
2004:0:5::/48                   bgp             200     198.51.100.2                ifl-0/1/31/10
2004:0:6::/48                   bgp             200     198.51.100.2                ifl-0/1/31/10
2004:0:7::/48                   bgp             200     198.51.100.2                ifl-0/1/31/10
2004:0:8::/48                   bgp             200     198.51.100.2                ifl-0/1/31/10
2004:0:9::/48                   bgp             200     198.51.100.2                ifl-0/1/31/10
2004:0:a::/48                   bgp             200     198.51.100.2                ifl-0/1/31/10
2004:0:b::/48                   bgp             200     198.51.100.2                ifl-0/1/31/10
2004:0:c::/48                   bgp             200     198.51.100.2                ifl-0/1/31/10
2004:0:d::/48                   bgp             200     198.51.100.2                ifl-0/1/31/10
2004:0:e::/48                   bgp             200     198.51.100.2                ifl-0/1/31/10
2004:0:f::/48                   bgp             200     198.51.100.2                ifl-0/1/31/10

<...>
```

Pinging an IPv6 route (source: bgp) from the Multiservice Edge.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> ping 2004:0:1::
68 bytes from 2004:0:1::: icmp_seq=1 ttl=253 time=10.0398 ms
68 bytes from 2004:0:1::: icmp_seq=2 ttl=253 time=2.9673 ms
68 bytes from 2004:0:1::: icmp_seq=3 ttl=253 time=6.2365 ms
68 bytes from 2004:0:1::: icmp_seq=4 ttl=253 time=7.9022 ms
68 bytes from 2004:0:1::: icmp_seq=5 ttl=253 time=1.5511 ms
Statistics: 5 sent, 5 received, 0% packet loss
```

1.4. PPPoE Subscriber Management Configuration

PPPoE (Point-to-Point Protocol over Ethernet) is a network protocol that provides connectivity across Ethernet networks. The protocol enables communication between network endpoints and service providers implement PPPoE to connect many hosts on a single Ethernet LAN to the core network through a common device. In RBFS, the PPPoE daemon (`pppoed`) manages PPPoE and PPP sessions.

For PPPoE Subscriber Management, the following configurations are mandatory:

1. Access Interface Configuration
2. Access Profile Configuration
3. AAA (Authentication, Authorization and Accounting) Profile Configuration. Based on the authentication requirement, configure any one of the following:
    a. Local Authentication
        i. Pool Configuration
        ii. User Profile Configuration
    b. RADIUS Authentication
        i. RADIUS Profile Configuration

ii. RADIUS Server Configuration

This solution section discusses RADIUS authentication.

NOTES:

- Access interfaces can be configured without VLAN tags (untagged) and with one (single tagged) or two (double tagged) VLAN tags.
- There can be more than one interface configured for subscriber management and each interface can reference the same profile.

1.4.1. Configuring PPPoE Subscriber Management

For detailed information about the subscriber configuration options, see the Subscriber Management Configuration Guide.

1. Configure the access profile `pppoe`.

```
set access access-profile pppoe
set access access-profile pppoe protocol pppoe enable true
set access access-profile pppoe protocol ppp lcp authentication-protocol PAP
set access access-profile pppoe protocol ppp lcp echo-interval 30
set access access-profile pppoe protocol ppp lcp echo-enable true
set access access-profile pppoe protocol ppp ipcp enable true
set access access-profile pppoe protocol ppp ipcp source-ifl lo-0/0/1/0
set access access-profile pppoe protocol ppp ip6cp enable true
set access access-profile pppoe protocol ra enable true
set access access-profile pppoe protocol ra interval 30
set access access-profile pppoe protocol dhcpv6 enable true
set access access-profile pppoe protocol dhcpv6 lifetime 14400
set access access-profile pppoe protocol dhcpv6 preferred-lifetime 1800
set access access-profile pppoe address-family ipv4 enable true
set access access-profile pppoe address-family ipv4 pool-name poolv4
set access access-profile pppoe address-family ipv4 primary-dns 203.0.113.200
set access access-profile pppoe address-family ipv4 secondary-dns 203.0.113.201
set access access-profile pppoe instance default
set access access-profile pppoe address-family ipv6 enable true
set access access-profile pppoe address-family ipv6 pool-name poolv6
set access access-profile pppoe address-family ipv6 prefix-delegation-pool-name
poolv6pd
set access access-profile pppoe address-family ipv6 primary-dns 2001:db8:0:20::1
set access access-profile pppoe address-family ipv6 secondary-dns 2001:db8:0:20::1
commit
```

The access profile configuration is shown below.

```
supervisor@rtbrick.net: cfg> show config access access-profile
{
  "rtbrick-config:access-profile": [
```

```
    {
      "profile-name": "pppoe",
      "protocol": {
        "pppoe": {
          "enable": "true"
        },
        "ppp": {
          "lcp": {
            "authentication-protocol": "PAP",
            "echo-interval": 30,
            "echo-enable": "true"
          },
          "ipcp": {
            "enable": "true",
            "source-ifl": "lo-0/0/1/0"
          },
          "ip6cp": {
            "enable": "true"
          }
        },
        "ra": {
          "enable": "true",
          "interval": 30
        },
        "dhcpv6": {
          "enable": "true",
          "lifetime": 14400,
          "preferred-lifetime": 1800
        }
      },
      "address-family": {
        "ipv4": {
          "enable": "true",
          "pool-name": "poolv4",
          "primary-dns": "203.0.113.200",
          "secondary-dns": "203.0.113.201",
          "instance": "default"
        },
        "ipv6": {
          "enable": "true",
          "pool-name": "poolv6",
          "prefix-delegation-pool-name": "poolv6pd",
          "primary-dns": "2001:db8:0:20::1",
          "secondary-dns": "2001:db8:0:20::1",
          "instance": "default"
        }
      }
    }
  ]
}
```

2. Configure the Authentication and Accounting (AAA) profile for `aaa-profile`.

```
set access aaa-profile aaa-profile
set access aaa-profile aaa-profile session-timeout 0
set access aaa-profile aaa-profile idle-timeout 0
set access aaa-profile aaa-profile aaa-radius-profile radius-profile
set access aaa-profile aaa-profile authentication order RADIUS
set access aaa-profile aaa-profile accounting order RADIUS
set access aaa-profile aaa-profile accounting interim-interval 86400
set access aaa-profile aaa-profile accounting ingress accounting-source POLICER
set access aaa-profile aaa-profile accounting egress accounting-source CLASS
commit
```

The access AAA configuration is shown below.

```
supervisor@rtbrick.net: cfg> show config access aaa-profile
{
  "rtbrick-config:aaa-profile": [
    {
      "profile-name": "aaa-profile",
      "session-timeout": 0,
      "idle-timeout": 0,
      "aaa-radius-profile": "radius-profile",
      "authentication": {
        "order": "RADIUS"
      },
      "accounting": {
        "order": "RADIUS",
        "interim-interval": 86400,
        "ingress": {
          "accounting-source": "POLICER"
        },
        "egress": {
          "accounting-source": "CLASS"
        }
      }
    }
  ]
}
```

3. Configure the access interface. Double-tagged interface is configured in this case as the access interface (`ifp-0/0/3`). The interface configuration assigns the access type, access profile, AAA profile, and further optional attributes like service-profile to the specified access interface.

```
set access interface double-tagged ifp-0/0/3 1001 1100 1001 1100
set access interface double-tagged ifp-0/0/3 1001 1100 1001 1100 access-type PPPoE
set access interface double-tagged ifp-0/0/3 1001 1100 1001 1100 access-profile-name
pppoe
```

```
set access interface double-tagged ifp-0/0/3 1001 1100 1001 1100 service-profile-name
qos_service
set access interface double-tagged ifp-0/0/3 1001 1100 1001 1100 aaa-profile-name aaa-
profile
commit
```

The double-tagged access interface configuration is shown below.

```
supervisor@rtbrick.net: cfg> show config access interface
{
  "rtbrick-config:interface": {
    "double-tagged": [
      {
        "interface-name": "ifp-0/0/3",
        "outer-vlan-min": 1001,
        "outer-vlan-max": 1100,
        "inner-vlan-min": 1001,
        "inner-vlan-max": 1100,
        "access-type": "PPPoE",
        "access-profile-name": "pppoe",
        "service-profile-name": "qos_service",
        "aaa-profile-name": "aaa-profile"
      }
    ]
  }
}
```

4. In this solution, we configure AAA authentication and accounting with RADIUS. To use RADIUS authentication and accounting both the RADIUS profile and RADIUS server configurations (see below) must be configured.

a. Configure RADIUS profile `radius-profile`.

```
set access radius-profile radius-profile
set access radius-profile radius-profile nas-ip-address 192.0.2.74
set access radius-profile radius-profile nas-port-format DEFAULT
set access radius-profile radius-profile nas-port-type Ethernet
set access radius-profile radius-profile authentication radius-server-profile-name
radius
set access radius-profile radius-profile accounting radius-server-profile-name radius
commit
```

The RADIUS profile configuration is shown below.

```
supervisor@rtbrick.net: cfg> show config access radius-profile radius-profile
{
  "rtbrick-config:radius-profile": [
```

```
    {
      "profile-name": "radius-profile",
      "nas-ip-address": "192.0.2.74",
      "nas-port-format": "DEFAULT",
      "nas-port-type": "Ethernet",
      "authentication": {
        "radius-server-profile-name": [
          "radius"
          ]
      },
      "accounting": {
        "radius-server-profile-name": [
          "radius"
          ]
      }
    }
  ]
}
```

b. Configure the RADIUS server `radius`.

```
set access radius-server radius
set access radius-server radius address 192.0.2.34
set access radius-server radius source-address 192.0.2.74
set access radius-server radius secret-encrypted-text
$2b2feb12f730107454b1be6a0f8242b0f
set access radius-server radius routing-instance default
set access radius-server radius rate 300
set access radius-server radius authentication enable true
set access radius-server radius authentication retry 3
set access radius-server radius authentication timeout 5
set access radius-server radius accounting enable true
set access radius-server radius accounting timeout 30
set access radius-server radius coa enable true
commit
```



The attribute `secret-plain-text` is converted to `secret-encrypted-text` in the show command output and the value is hashed.

The RADIUS server configuration is shown below.

```
supervisor@rtbrick.net: cfg> show config access radius-server radius
{
  "rtbrick-config:radius-server": [
    {
      "server-name": "radius",
      "address": "192.0.2.34",
      "source-address": "192.0.2.74",
```

```
      "secret-encrypted-text": "$2b2feb12f730107454b1be6a0f8242b0f",
      "routing-instance": "default",
      "rate": 300,
      "authentication": {
        "enable": "true",
        "retry": 3,
        "timeout": 5
      },
      "accounting": {
        "enable": "true",
        "timeout": 30
      },
      "coa": {
        "enable": "true"
      }
    }
  ]
}
```

5. Configure the IPv4 and IPv6 access pools.

```
set access pool poolv4
set access pool poolv4 ipv4-address low 203.0.113.1
set access pool poolv4 ipv4-address high 203.0.113.64
set access pool poolv6
set access pool poolv6 ipv6-prefix low 2001:db8:0:1::/64
set access pool poolv6 ipv6-prefix high 2001:db8:0:2::/64
set access pool poolv6pd
set access pool poolv6pd ipv6-prefix low fc56:100:1::/56
set access pool poolv6pd ipv6-prefix high fc56:100:5000::/56
commit
```

The access pool configuration is shown below.

```
supervisor@rtbrick.net: cfg> show config access pool
{
  "rtbrick-config:pool": [
    {
      "pool-name": "poolv4",
      "ipv4-address": {
        "low": "203.0.113.1",
        "high": "203.0.113.64"
      }
    },
    {
      "pool-name": "poolv6",
      "ipv6-prefix": {
        "low": "2001:db8:0:1::/64",
        "high": "2001:db8:0:2::/64"
```

```
      }
    },
    {
      "pool-name": "poolv6pd",
      "ipv6-prefix": {
        "low": "fc56:100:1::/56",
        "high": "fc56:100:5000::/56"
      }
    }
  ]
}
```

1.4.2. PPPoE Quality of Service (QoS) Configuration



The QoS model explained in this document uses a complex HQoS model with the intent to showcase the complete range of QoS features available in RBFS. However, it may not be needed or desirable for all deployments. In such a case it should be possible to conceive of a simple QoS model as required by simplifying the provided QoS model.

Following are the steps involved in configuring and verifying PPPoE QoS:

1. Configuring service profile to enable QoS on PPPoE subscriber
2. Configuring downstream QoS
3. Configuring upstream QoS
4. Configuring QoS remarking
5. Configuring PPPoE subscriber accounting for upstream and downstream traffic
6. Configuring PPPoE subscribers QoS on BNG Blaster
7. Validating PPPoE QoS on BNG Blaster

The figure below shows how QoS is configured for ingress and egress traffic.

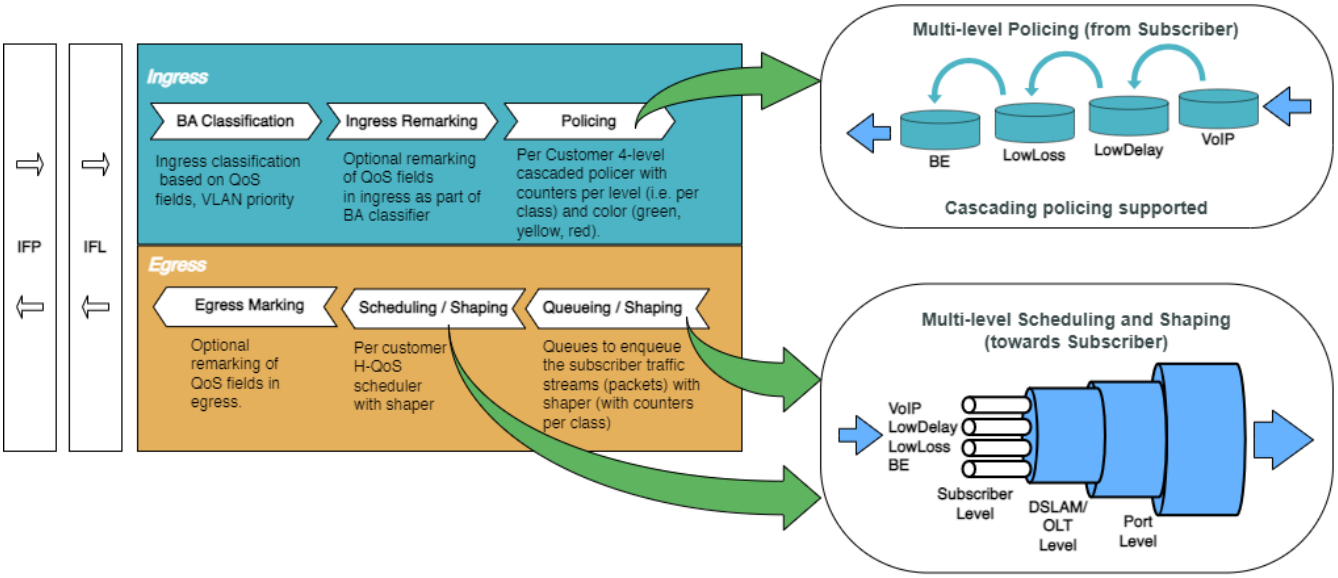


Figure 3. Hierarchical Quality of Service Primitives

For detailed information about the QoS configuration options, see HQoS Configuration Guide.

Configure Service Profile

Service profile configuration in subscriber management allows to assign QoS configurations to a subscriber.

1. Configure the service profile to enable QoS. The service profile defined to enable Quality of Service with the profile name is `residential`.

```
set access service-profile qos_service qos profile residential
commit
```

The configuration of the service profile named `residential` is shown below.

```
supervisor@rtbrick.net: cfg> show config access service-profile qos_service
{
  "rtbrick-config:service-profile": [
    {
      "profile-name": "qos_service",
      "qos": {
        "profile": "residential"
      }
    }
  ]
}
```

2. Enable QoS on PPPoE subscriber access interface (`ifp-0/0/3`) to enable QoS for PPPoE subscriber.

```
set access interface double-tagged ifp-0/0/3 1001 1100 1001 1100 service-profile-name
```

```
qos_service
commit
```

Below is the double-tagged access interface on which the service profile `qos_service` is configured.

```
supervisor@rtbrick.net: cfg> show config access interface double-tagged ifp-0/0/3
{
  "rtbrick-config:double-tagged": [
    {
      "interface-name": "ifp-0/0/3",
      "outer-vlan-min": 1001,
      "outer-vlan-max": 1100,
      "inner-vlan-min": 1001,
      "inner-vlan-max": 1100,
      "access-type": "PPPoE",
      "service-profile-name": "qos_service",
      "aaa-profile-name": "aaa-profile",
      "gateway-ifl": "lo-0/0/1/0"
    }
  ]
}
```

3. Configure QoS profile to enable on PPPoE subscriber.

```
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 class-policer-map-name
policer-map-residential
set forwarding-options class-of-service profile residential policer-name policer-
residential
set forwarding-options class-of-service profile residential scheduler-map-name subs-
4queues-residential
commit
```

The QoS Profile with all the primitives needed to enable traffic profiles on PPPoE Subscribers is as follows:

```
supervisor@rtbrick.net: cfg> show config forwarding-options class-of-service profile
reside
ntial
{
  "rtbrick-config: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"
    }
  ]
}
```

Configure Downstream QoS

Downstream Quality of Service (QoS) is used to prioritize network traffic from the Internet to subscribers.

1. Enable global classification for downstream traffic.

```
set forwarding-options class-of-service global multifield-classifier-name global_mfc
commit
```

Below is the multi-field-classifier (MFC) based classifier for global enabling of downstream traffic classification.

```
supervisor@rtbrick.net: cfg> show config forwarding-options class-of-service global
multifield-classifier-name
{
  "rtbrick-config:multifield-classifier-name": "global_mfc"
}
```

2. Configure the MFC-based classifier with qualifiers and actions.

```
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 192.0.2.2/32
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 192.0.2.2/32
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1002 action forward-class class-1
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1003
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1003 match ipv4-tos 192
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1003 match source-ipv4-prefix 192.0.2.2/32
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1003 action forward-class class-2
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1004
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1004 match ipv4-tos 224
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1004 match source-ipv4-prefix 192.0.2.2/32
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1004 action forward-class class-3
commit
```

The configuration of the QoS MFC-based Classifier for the classification of downstream traffic from the core towards PPPoE Subscriber is shown below.

```
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": "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"
          }
        }
      ]
    }
  ]
}
```

3. Enqueue classified traffic to different queues using class-to-queue mapping.

```
set forwarding-options class-of-service queue-group subs-4queues queue-numbers 4
set forwarding-options class-of-service class-queue-map subs-4queues
set forwarding-options class-of-service class-queue-map subs-4queues class class-0
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
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
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
set forwarding-options class-of-service class-queue-map subs-4queues class class-3
queue-name VO_SUBS
commit
```

Below is the QoS class-queue mapping configuration:

```
supervisor@rtbrick.net: cfg> show config forwarding-options class-of-service class-
queue-map subs-4queues class
{
```

```
  "rtbrick-config: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"
    }
  ]
}
```

4. Configure the queues needed for enqueuing and dequeuing traffic streams.

```
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 shaper-name shaper_VO
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
commit
```

The queue Configuration is shown below.

```
supervisor@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"
      }
    }
  ]
}
```

5. Configure the scheduler needed by Subscriber/Session scheduler-map and OLT scheduler-map.

```
set forwarding-options class-of-service scheduler pon0
set forwarding-options class-of-service scheduler pon0 type fair_queueing
set forwarding-options class-of-service scheduler pon0 shaper-name gpon-shaper
set forwarding-options class-of-service scheduler subs-4queues
set forwarding-options class-of-service scheduler subs-4queues shaper-name
shaper_session
set forwarding-options class-of-service scheduler subs-4queues type strict_priority
set forwarding-options class-of-service scheduler subs-4queues composite false
commit
```

The configuration of the scheduler-map and OLT scheduler-map is shown below.

```
supervisor@rtbrick.net: cfg> show config  forwarding-options class-of-service
scheduler
{
  "rtbrick-config:scheduler": [
    {
      "scheduler-name": "pon0",
      "shaper-name": "gpon-shaper",
      "type": "fair_queueing"
    },
    {
      "scheduler-name": "subs-4queues",
      "shaper-name": "shaper_session",
      "type": "strict_priority",
      "composite": "false"
    }
  ]
}
```

6. Configure the session/subscriber scheduler mapping for dequeuing traffic based on scheduler type for each queue:

```
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 QoS Subscriber/Session Scheduler-Map configuration is shown below:

```
supervisor@rtbrick.net: cfg> show config forwarding-options class-of-service
scheduler-map subs-4queues-residential
{
  "rtbrick-config:scheduler-map": [
    {
      "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"
        }
      ]
    }
  ]
}
```

7. Configure the OLT scheduler-mapping for each PON to be scheduled according to the scheduler type.

```
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
commit
```

The OLT Scheduler-Map configuration is shown below:

```
supervisor@rtbrick.net: cfg> show config forwarding-options class-of-service
scheduler-map schedmap-olt
{
  "rtbrick-config:scheduler-map": [
    {
      "scheduler-map-name": "schedmap-olt",
      "scheduler-name": [
        {
          "name": "pon0",
          "port-connection": "scheduler_to_port"
        }
      ]
    }
  ]
}
```

8. Configure downstream traffic shaping for both session schedulers and queues.



Queue Shaping is only on VO_SUBS Queue.

```
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
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 gpon-shaper
set forwarding-options class-of-service shaper gpon-shaper shaping-rate-high 2488000
set forwarding-options class-of-service shaper gpon-shaper shaping-rate-low 32000
commit
```

The shaping Configuration is shown below.

```
supervisor@rtbrick.net: cfg> show config forwarding-options class-of-service shaper
{
  "rtbrick-config:shaper": [
    {
      "shaper-name": "gpon-shaper",
      "shaping-rate-high": 2488000,
      "shaping-rate-low": 32000
    },
    {
      "shaper-name": "shaper_VO",
      "shaping-rate-high": 2000,
      "shaping-rate-low": 0
    },
    {
      "shaper-name": "shaper_session",
      "shaping-rate-high": 10000,
      "shaping-rate-low": 100
    }
  ]
}
```

Configure Upstream QoS

1. Configure the BA Classifier for the classification of multiple traffic streams targeted at PPPoE subscribers:

```
set forwarding-options class-of-service classifier subs-pbit-class
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 1
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
```

```
802.1 codepoint 1 class class-0
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 1 remark-codepoint 7
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 2
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 2 class class-1
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 2 remark-codepoint 7
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 3
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 3 class class-2
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 3 remark-codepoint 7
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 4
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 4 class class-3
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 4 remark-codepoint 7
commit
```

The configuration of the QoS BA-based Classifier for the classification of upstream traffic toward the PPPoE Subscriber is shown below.

```
supervisor@rtbrick.net: cfg> show config forwarding-options class-of-service
classifier subs-pbit-class
{
  "rtbrick-config: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
            }
          ]
        }
      ]
    }
  ]
}
```

2. Configure multi-level policer to police 4-Level traffic.

```
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 multi-level policer configuration is shown below:

```
supervisor@rtbrick.net: 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"
    }
  ]
}
```

3. Map the classified traffic streams to different policer levels using class-to-policer mapping:

```
set forwarding-options class-of-service class-policer-map policer-map-residential
set forwarding-options class-of-service class-policer-map policer-map-residential
class class-0
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
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
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
set forwarding-options class-of-service class-policer-map policer-map-residential
class class-3 policer-level level-4
commit
```

The class-policer-map configuration is shown below:

```
supervisor@rtbrick.net: cfg> show config forwarding-options class-of-service class-
policer-map policer-map-residential class
{
  "rtbrick-config: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"
    }
  ]
}
```

Configure QoS Remarking

1. Remark downstream traffic egressing from subscriber interface (egress remarking).

```
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 1
set forwarding-options class-of-service remark-map subs-remarking-residential remark-
type ieee-802.1 match-codepoint 1 color all
```

```
set forwarding-options class-of-service remark-map subs-remarking-residential remark-
type ieee-802.1 match-codepoint 1 color all remark-codepoint 6
set forwarding-options class-of-service remark-map subs-remarking-residential remark-
type ieee-802.1 match-codepoint 2
set forwarding-options class-of-service remark-map subs-remarking-residential remark-
type ieee-802.1 match-codepoint 2 color all
set forwarding-options class-of-service remark-map subs-remarking-residential remark-
type ieee-802.1 match-codepoint 2 color all remark-codepoint 6
set forwarding-options class-of-service remark-map subs-remarking-residential remark-
type ieee-802.1 match-codepoint 3
set forwarding-options class-of-service remark-map subs-remarking-residential remark-
type ieee-802.1 match-codepoint 3 color all
set forwarding-options class-of-service remark-map subs-remarking-residential remark-
type ieee-802.1 match-codepoint 3 color all remark-codepoint 6
set forwarding-options class-of-service remark-map subs-remarking-residential remark-
type ieee-802.1 match-codepoint 4
set forwarding-options class-of-service remark-map subs-remarking-residential remark-
type ieee-802.1 match-codepoint 4 color all
set forwarding-options class-of-service remark-map subs-remarking-residential remark-
type ieee-802.1 match-codepoint 4 color all remark-codepoint 6
commit
```

The remarking configuration is shown below:

```
supervisor@rtbrick.net: cfg> show config forwarding-options class-of-service remark-
map subs-remarking-residential
{
  "rtbrick-config:remark-map": [
    {
      "remark-map-name": "subs-remarking-residential",
      "remark-type": [
        {
          "remark-type": "ieee-802.1",
          "match-codepoint": [
            {
              "match-codepoint": 1,
              "color": [
                {
                  "color": "all",
                  "remark-codepoint": 6
                }
              ]
            },
            {
              "match-codepoint": 2,
              "color": [
                {
                  "color": "all",
                  "remark-codepoint": 6
                }
```

```
            ]
          },
          {
            "match-codepoint": 3,
            "color": [
              {
                "color": "all",
                "remark-codepoint": 6
              }
            ]
          },
          {
            "match-codepoint": 4,
            "color": [
              {
                "color": "all",
                "remark-codepoint": 6
              }
            ]
          }
        ]
      }
    ]
  }
]
}
```

3. Remark upstream traffic ingressing to a subscriber's interface [ingress remarking]



In the upstream traffic classifier configuration shown below, remarking of all traffic streams with code point '7' is done.

```
set forwarding-options class-of-service classifier subs-pbit-class
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 1
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 1 class class-0
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 1 remark-codepoint 7
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 2
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 2 class class-1
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 2 remark-codepoint 7
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 3
```

```
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 3 class class-2
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 3 remark-codepoint 7
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 4
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 4 class class-3
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 4 remark-codepoint 7
commit
```

Below is the remarking configuration:

```
supervisor@rtbrick.net: cfg> show config forwarding-options class-of-service
classifier subs-pbit-class
{
  "rtbrick-config: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
            }
          ]
        }
      ]
    }
  ]
```

```
}
```

1.4.3. Configure Mirroring for Downstream Traffic Remark Validation

Configure Mirroring for Downstream Traffic

```
set forwarding-options mirror m1
set forwarding-options mirror m1 destination interface cpu-0/0/200
set forwarding-options mirror m1 source direction egress
set forwarding-options mirror m1 source interface ifp-0/0/3
commit
```

Validate the downstream traffic remarking



This validation requires mirroring the subscriber access interface on the Multiservice Edge device.

```
supervisor@rtbrick.net: cfg> show config forwarding-options mirror
{
  "rtbrick-config:mirror": [
    {
      "name": "m1",
      "destination": {
        "interface": "cpu-0/0/200"
      },
      "source": {
        "direction": "egress",
        "interface": "ifp-0/0/3"
      }
    }
  ]
}
```

The capture mirroring can be performed as shown below.

```
supervisor@rtbrick.net: cfg> capture mirror

2023-10-17T09:35:41.690237+0000 e8:c5:7a:b5:d2:a5 > 01:80:c2:00:00:0e, ethertype LLDP
(0x88cc), length 121: LLDP, length 107
        Chassis ID TLV (1), length 7
          Subtype MAC address (4): e8:c5:7a:b5:d2:9c
        Port ID TLV (2), length 11
          Subtype Interface Name (5): ifp-0/0/3
        Time to Live TLV (3), length 2: TTL 121s
        Port Description TLV (4), length 42: Physical interface #1 from node 0, chip 0
        System Name TLV (5), length 33: ufi11.q2a.u41.r3.nbg.rtbrick.net
```

```
        End TLV (0), length 0

2023-10-17T09:36:11.689298+0000 e8:c5:7a:b5:d2:a5 > 01:80:c2:00:00:0e, ethertype LLDP
(0x88cc), length 121: LLDP, length 107
        Chassis ID TLV (1), length 7
          Subtype MAC address (4): e8:c5:7a:b5:d2:9c
        Port ID TLV (2), length 11
          Subtype Interface Name (5): ifp-0/0/3
        Time to Live TLV (3), length 2: TTL 121s
        Port Description TLV (4), length 42: Physical interface #1 from node 0, chip 0
        System Name TLV (5), length 33: ufi11.q2a.u41.r3.nbg.rtbrick.net
        End TLV (0), length 0
```

Configure Mirroring for Upstream Traffic Remark Validation

```
set forwarding-options mirror m1
set forwarding-options mirror m1 destination interface cpu-0/0/200
set forwarding-options mirror m1 source direction egress
set forwarding-options mirror m1 source interface ifp-0/0/3
commit
```

Validating the upstream traffic remarking



Mirror the core-facing port on the Multiservice Edge device as shown below.

```
supervisor@rtbrick.net: cfg> show config forwarding-options mirror m1
{
  "rtbrick-config:mirror": [
    {
      "name": "m1",
      "destination": {
        "interface": "cpu-0/0/200"
      },
      "source": {
        "direction": "ingress",
        "interface": "ifp-0/0/3"
      }
    }
  ]
}
```

The capture mirroring can be performed as shown below. It confirms all four traffic streams noted with codepoint=7.

```
supervisor@rtbrick.net: cfg> capture mirror

2023-04-03T13:53:31.400011+0000 02:00:00:00:00:01 > e8:c5:7a:8f:76:f1, ethertype
```

```
802.1Q (0x8100), length 1022: vlan 1001, p 2, ethertype 802.1Q, vlan 1001, p 2,
ethertype IPv4, (tos 0xa0, ttl 64, id 0, offset 0, flags [DF], proto UDP (17), length
1000)
    203.0.113.1.65056 > 192.0.2.2.65056: UDP, length 972

2023-04-03T13:53:31.400167+0000 02:00:00:00:00:01 > e8:c5:7a:8f:76:f1, ethertype
802.1Q (0x8100), length 1022: vlan 1001, p 3, ethertype 802.1Q, vlan 1001, p 3,
ethertype IPv4, (tos 0xc0, ttl 64, id 0, offset 0, flags [DF], proto UDP (17), length
1000)
    203.0.113.1.65056 > 192.0.2.2.65056: UDP, length 972
```

1.4.4. Configure FreeRADIUS Server

Installation of FreeRADIUS

FreeRADIUS server can be installed on any Linux OS distribution. For information about installing FreeRADIUS, see Installing the FreeRADIUS Server.

Remove the Unsupported Files

It is necessary to remove `echo`, `ntlm_auth`, `eap`, `echo`, and `mschap` files once FreeRadius has been installed, since they are not required by this reference design. To remove these files, enter the following commands:

```
rm -rf /etc/freeradius/3.0/mods-enabled/echo
rm -rf /etc/freeradius/3.0/mods-enabled/ntlm_auth
rm -rf /etc/freeradius/3.0/mods-enabled/eap
rm -rf /etc/freeradius/3.0/mods-enabled/echo
rm -rf /etc/freeradius/3.0/mods-enabled/mschap
```

FreeRADIUS User and Group Settings

Using the following commands, one can set the user or group as root.

```
sed -i 's/User=freerad/User=root/g' /lib/systemd/system/freeradius.service
sed -i 's/Group=freerad/Group=root/g' /lib/systemd/system/freeradius.service
```

Run the command for reloading 'systemd' after user or group settings:

```
systemctl daemon-reload
```

Configure the FreeRADIUS Files

It is necessary to configure the FreeRADIUS files once FreeRadius has been installed. The configuration files can be found under `/etc/freeradius/3.0/`.

authorize

Using the following command, one can view the `authorize` file in its default location.

```
~:/etc/freeradius/3.0 # cat mods-config/files/authorize
```

Replace the content of the `authorize` file with the following:

```
$INCLUDE /etc/freeradius/3.0/pppoe_users_file
```



This file can also be downloaded from the appendix (Appendix C: RADIUS Server Configuration).

pppoe_users_file

The PPPoE Users file (`pppoe_users_file`) mentioned in the above section includes subscriber profile parameters as shown below.

```
"02:00:00:00:00:01@pppoe" Cleartext-Password := "pppoe"
    Service-Type = Framed-User,
    Class = PPPOE,
    Framed-IP-Address = 203.0.113.1,
    Framed-IP-Netmask = 255.255.255.255,
    RtBrick-DNS-Primary = 203.0.113.200,
    RtBrick-DNS-Secondary  = 203.0.113.201,
    Framed-IPv6-Prefix = 2001:db8:0:1::1/64,
    Delegated-IPv6-Prefix = 2001:db8:0:3::/56,
    RtBrick-DNS-Primary-IPv6  = 2001:db8:0:20::1,
    RtBrick-DNS-Secondary-IPv6 = 2001:db8:0:20::2,
    Session-Timeout = 0,
    Idle-Timeout = 0,
    Reply-Message = "FOOBAR Internet",
    Acct-Interim-Interval = 120,
    RtBrick-QoS-Profile = "residential"

DEFAULT User-Name =~ '^[0-9a-f\:]+@pppoe$', Cleartext-Password := 'pppoe'
    Service-Type = Framed-User,
    Class = "PPPOE",
    Acct-Interim-Interval = 900
```

The `pppoe_users_file` can be created with the above content in the `/etc/freeradius/3.0/` path. Alternatively, this file can be downloaded from the appendix section of this guide and placed at `/etc/freeradius/3.0/`.

clients.conf

Clients.conf file shall be configured with the expected RADIUS client IP address and secret.

```
~:/etc/freeradius/3.0 # cat clients.conf
client rtbrick {
        ipaddr          = 192.0.2.74
        secret          = testing123
        shortname       = rtbrick
        nas_type        = other
        require_message_authenticator = no
}
```

The `clients.conf` file (/etc/freeradius/3.0/clients.conf) used for this reference design can be downloaded from the appendix section of this guide.

radiusd.conf

The `radiusd.conf` file should be configured with the expected RADIUS authentication and accounting parameters.

```
prefix = /usr
exec_prefix = /usr
sysconfdir = /etc
localstatedir = /var
sbindir = ${exec_prefix}/sbin
logdir = /var/log/freeradius
raddbdir = /etc/freeradius
radacctdir = ${logdir}/radacct
name = freeradius
confdir = ${raddbdir}
modconfdir = ${confdir}/mods-config
run_dir = ${localstatedir}/run/${name}
db_dir = ${raddbdir}
libdir = /usr/lib/freeradius
pidfile = ${run_dir}/${name}.pid
correct_escapes = true
max_request_time = 5
cleanup_delay = 0
max_requests = 16384
hostname_lookups = no
log {
    destination = files
    file = ${logdir}/radius.log
    stripped_names = no
    auth = yes
}
checkrad = ${sbindir}/checkrad
```

```
security {
    # user = radius
    # group = radius
    allow_core_dumps = no
    max_attributes = 200
    reject_delay = 0
    status_server = no
@openssl_version_check_config@
}
proxy_requests  = no
$INCLUDE clients.conf
thread pool {
    start_servers = 32
    max_servers = 128
    min_spare_servers = 8
    max_spare_servers = 16
    max_queue_size = 16384
    max_requests_per_server = 0
    auto_limit_acct = no
}
modules {
    $INCLUDE mods-enabled/
}
instantiate {
    files
    linelog
}
server default {
    listen {
        type = auth
        ipaddr = *
        port = 1812
    }
    listen {
        type = acct
        ipaddr = *
        port = 1813
    }
    authorize {
        update request {
            FreeRADIUS-Client-Shortname = "%{&request:Client-Shortname}"
        }
        if (&request:Client-Shortname == "rtbrick-server") {
            rtbrick-server-log
        }
        files
        pap
    }
    authenticate {
        pap
    }
```

```
    post-auth {
        if (&request:Client-Shortname == "rtbrick-server") {
            rtbrick-server-log
        }
        Post-Auth-Type REJECT {
            update reply {
                Reply-Message := "Login Failed. Please check your username and
password."
            }
            attr_filter.access_reject
        }
    }
    preacct {
        ok
    }
    accounting {
        if (&request:Client-Shortname == "rtbrick-server") {
            rtbrick-server-log
        }
    ok
    }
    session {
    }
}
```

Radiusd.conf should be configured to use UDP ports 1812 and 1813 for authentication and accounting, respectively. Additionally, `rtbrick-server-log` should be added to the parameters for authorise, authenticate, and accounting.

The `radiusd.conf` file (/etc/freeradius/3.0/radiusd.conf) used for this reference design can be downloaded from the appendix section (Appendix C) of this guide.

detail

The `detail` file shall be configured for the RADIUS accounting logs.

```
/etc/freeradius/3.0 # cat mods-enabled/detail
    permissions = 0666detail rtbrick-server-log {
    filename = ${radacctdir}/rtbrick-server-detail.log

    header = "%t;%{NAS-IP-Address};%I;%{Packet-Src-Port}"
    log_packet_header = no
}
```

Ensure that `rtbrick-server-log` is specified in the `detail` file.

The `detail` file (/etc/freeradius/3.0/mods-enabled/detail) used for this reference design can be downloaded from the appendix section (Appendix C) of this guide.

dictionary.rtbrick

Add the RtBrick RADIUS dictionary (dictionary.rtbrick) to `/usr/share/freeradius/dictionary.rtbrick` and include it in ` /usr/share/freeradius/dictionary `.

The `dictionary.rtbrick` contains the RBFS attributes in FreeRADIUS format.

➔ Click here to download the `radius_config.zip`, which contains the `dictionary.rtbrick` file.

Stopping and Starting the FreeRADIUS Server for any Changes

For any changes, stop and restart the FreeRADIUS server.

To stop the server, enter the following command:

```
sudo service freeradius stop
```

To start the server, enter the following command:

```
sudo service freeradius start
```

The FreeRadius server is now ready to provide AAA (Authentication, Accounting & Authorization) services to logging in subscribers.

Displaying the RADIUS Service Status

Run the following command to determine whether RADIUS service is active:

```
sudo service freeradius status
```

1.4.5. Validating PPPoE Subscriber Bring-Up

Using traffic streams on both upstream and downstream directions with traffic packets and bytes statistics, PPPoE Subscriber sessions can be "ESTABLISHED".

The validation can be performed in two steps:

1. Establishing the PPPoE subscriber
2. Pinging the subscriber IPv4/IPv6 address

BNG Blaster - PPPoE Subscribers with Traffic Streams

Using BNG Blaster, which emulates PPPoE clients, session traffic ("session-traffic") and streams traffic with different code-points ("streams"), one can test PPPoE subscriber management feature.

The data traffic can be defined in Blaster by configuring session/streams traffic.

- Session Traffic in Blaster can be enabled by specifying `"autostart": true`. Once the session is established, traffic starts automatically. Eventually if you want to stop or start the traffic (session or streams), press F7/F8.
- Streams traffic in Blaster can be enabled by specifying `stream-group-id` at both streams (`"streams":`) and access interface levels (`interfaces:access`).

For information about using BNG Blaster, see Starting BNG Blaster.

Validating the PPPoE Session on BNG Blaster

To validate the PPPoE Session on BNG Blaster, switch to the Service Node. On the BNG Blaster terminal, the count of the "Established" session will be 1. Also, the logging of the same terminal will display "ALL SESSIONS ESTABLISHED" and "ALL SESSION TRAFFIC FLOWS VERIFIED" as highlighted in the below image.

In the image below, one can see the details of the established Single Subscriber session.

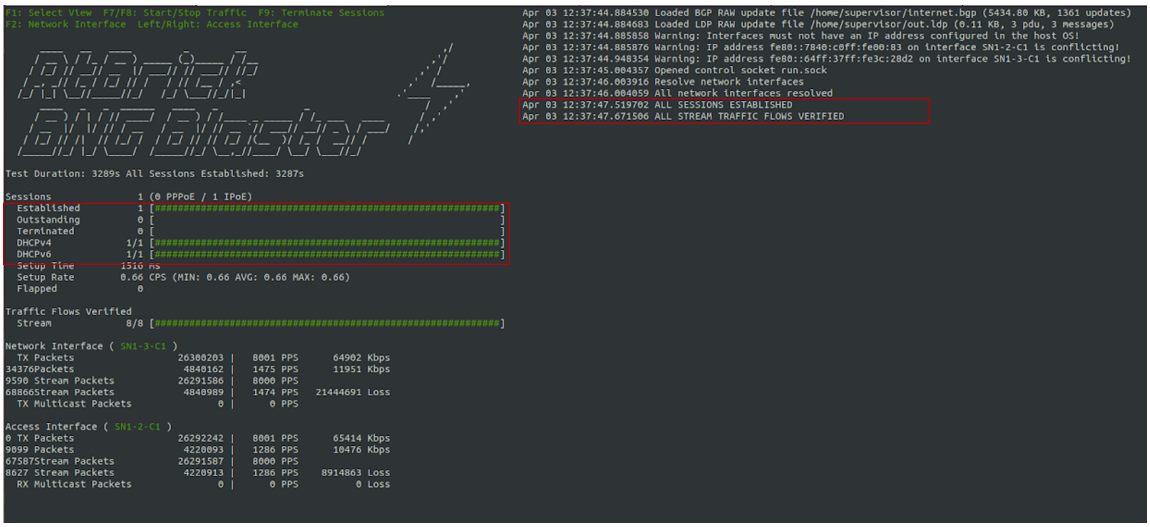


Figure 4. BNG Blaster Terminal View

Visit the following URL for more information on BNG Blaster:

https://rtbrick.github.io/bngblaster/

Viewing the Subscribers and the Subscriber Details on RBFS

Enter the following command to view the list of subscribers.

```
supervisor@rtbrick.net: cfg> show subscriber
Subscriber-Id          Interface        VLAN      Type   State
216454257090494470     ifp-0/0/3       1001:1001 PPPoE   ESTABLISHED
```

Enter the following command to view the details of the subscriber with ID `216454257090494470`.

```
supervisor@rtbrick.net: cfg> show subscriber 216454257090494470 detail
Subscriber-Id: 216454257090494470
    Type: PPPoE
```

```
    State: ESTABLISHED
    Created: Mon Apr 03 13:51:17 GMT +0000 2023
    Interface: ifp-0/0/3
    Outer VLAN: 1001
    Inner VLAN: 1001
    Client MAC: 02:00:00:00:00:01
    Server MAC: e8:c5:7a:8f:76:f1
    IFL: pppoe-0/1/30/216454257090494470
    Username: 02:00:00:00:00:01@pppoe
    Access-Profile: pppoe
    AAA-Profile: aaa-profile
    Service-Profile: qos_service
    Reply-Message: FOOBAR Internet
    Session-Timeout: 0 (disabled)
    Idle-Timeout: 0 (disabled)
    MTU: 1500 Profile: N/A
    IPv4:
        Instance: default
        Address: 203.0.113.1/255.255.255.255
        Address Active: True
        Primary DNS: 203.0.113.200
        Secondary DNS: 203.0.113.201
    IPv6:
        Instance: default
        RA Prefix: 2001:db8:0:1::1/64
        RA Prefix Active: True
        Delegated Prefix (DHCPv6): 2001:db8:0:3::/56
        Delegated Prefix Active: True
        Primary DNS: 2001:db8:0:20::1
        Secondary DNS: 2001:db8:0:20::2
    Accounting:
        Session-Id: 216454257090494470:1680529877
        Start-Time: 2023-04-03T13:51:18.543868+0000
        Interims Interval: 120 seconds
```

Pinging the Subscriber (source: PPPoE) from Multiservice Edge.

Before pinging a subscriber, use the `show route <···>` command to display the subscriber IPs at the Multiservice Edge.

```
supervisor@rtbrick.net: cfg> show route ipv4 unicast source pppoe
Instance: default, AFI: ipv4, SAFI: unicast
Prefix/Label                            Source            Pref    Next Hop
Interface
203.0.113.1/32                          pppoe              7       -
pppoe-0/1/30/216454257090494470
```

```
supervisor@rtbrick.net: cfg> show route ipv6 unicast source pppoe
Instance: default, AFI: ipv6, SAFI: unicast
```

```
Prefix/Label                              Source            Pref    Next Hop
Interface
2001:db8:0:1::1/64                        pppoe              7       -
pppoe-0/1/30/216454257090494470
2001:db8:0:3::/56                         pppoe              7
fe80::ffff:ffff:ff00:1                    pppoe-0/1/30/216454257090494470
```

From the above list, ping `203.0.113.1` as shown below.

```
supervisor@rtbrick.net: cfg> ping 203.0.113.1
68 bytes from 203.0.113.1: icmp_seq=1 ttl=64 time=1.3463 ms
68 bytes from 203.0.113.1: icmp_seq=2 ttl=64 time=10.5234 ms
68 bytes from 203.0.113.1: icmp_seq=3 ttl=64 time=9.8053 ms
68 bytes from 203.0.113.1: icmp_seq=4 ttl=64 time=11.0041 ms
68 bytes from 203.0.113.1: icmp_seq=5 ttl=64 time=10.9300 ms
Statistics: 5 sent, 5 received, 0% packet loss
```

Validating Traffic Streams

Traffic streams can be used to perform various forwarding verifications.

For upstream traffic capture, enter the following command:

```
capture interface ifp-0/0/3 direction in
```

For downstream traffic capture, enter the following command:

```
capture interface ifp-0/0/3 direction out
```

Here, `ifp-0/0/3` refers to the access interface.

Validating the PPPoE QoS on BNG Blaster

To validate the PPPoE QoS on BNG Blaster, switch to the Service Node. Navigate to the Streams and Session Traffic terminal by pressing F1 function key on your keyboard.

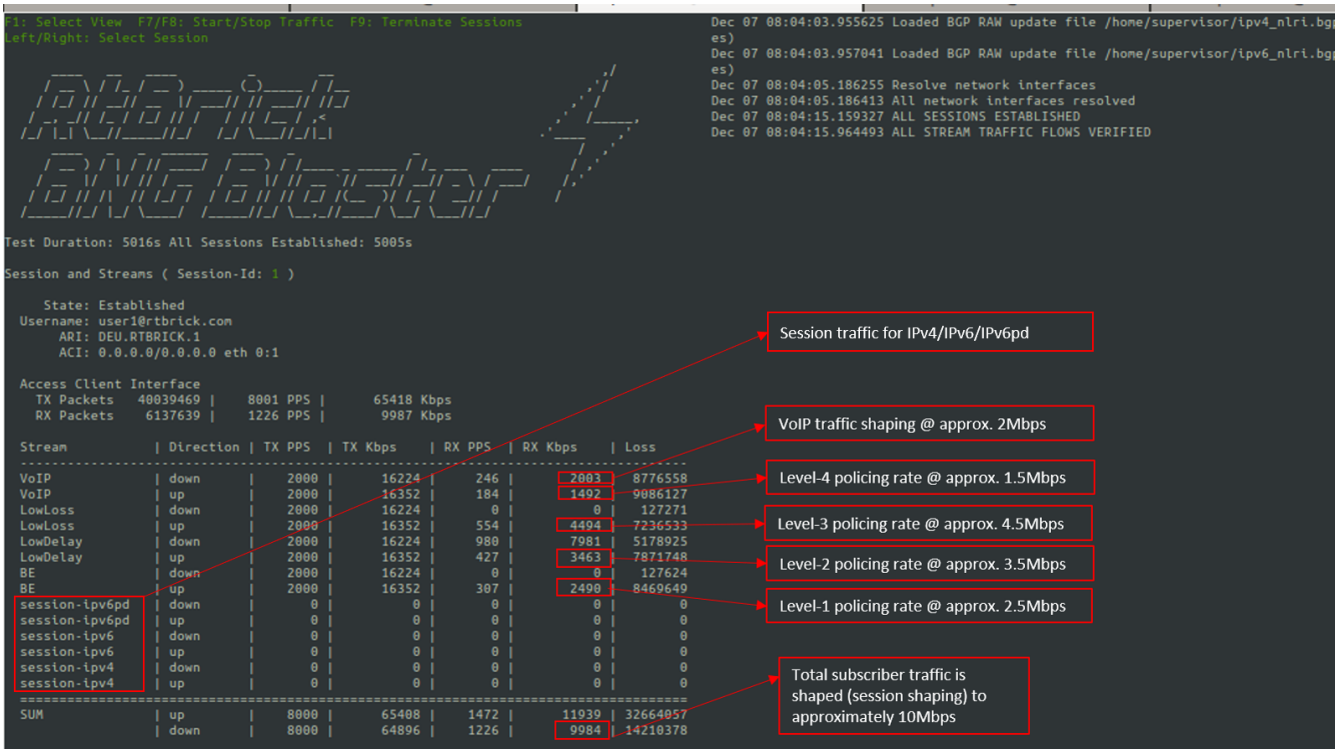


Figure 5. Reading output from BNG Blaster

As shown in the above image, the VoIP downstream traffic has been shaped (session shaping) to 2Mbps. Similarly, the total subscriber traffic has been shaped approximately to 10Mbps.

Following are the upstream traffic rates of different policer levels:

- Level-1 Rates ~=2.5Mbps
- Level-2 Rates ~=3.5Mbps
- Level-3 Rates ~=4.5Mbps
- Level-4 Rates ~=1.5Mbps

1.4.6. PPPoE Subscriber Accounting for Upstream and Downstream Traffic

Run the "show subscriber" command to view the list of subscribers.

```
supervisor@rtbrick.net: cfg> show subscriber
Subscriber-Id          Interface        VLAN      Type   State
72339069014638969      ifp-0/0/3        1001:1001 PPPoE  ESTABLISHED
```

Specify the Subscriber-ID to find the specific subscriber’s accounting details:

```
supervisor@rtbrick.net: cfg> show subscriber 72339069014638973  accounting
Subscriber-Id: 72339069014638973
    IFL: ppp-0/0/3/72339069014638973
    Start Timestamp: Wed Oct 18 04:39:37 GMT +0000 2023
    Idle Timestamp: Wed Oct 18 04:41:57 GMT +0000 2023
    Session-Timeout: 0 seconds
    Idle-Timeout: 0 seconds
```

```
    Session Statistics:
        Ingress: 201991 packets 202796060 bytes
        Egress: 178122 packets 183465660 bytes
    LIF Statistics:
        Ingress: 0 packets 0 bytes
        Egress: 0 packets 0 bytes
    Egress Class (Queue) Statistics:
        class-0: 309 packets 318270 bytes dropped: 271520 packets 272606080 bytes
        class-1: 142777 packets 147060310 bytes dropped: 133405 packets 133938620
bytes
        class-2: 307 packets 316210 bytes dropped: 268281 packets 269354124 bytes
        class-3: 34729 packets 35770870 bytes dropped: 241925 packets 242892700 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: 42088 packets 42253448 bytes dropped: 234461 packets 235394004 bytes
        Level 2: 58943 packets 59178772 bytes dropped: 217508 packets 218378032 bytes
        Level 3: 75566 packets 75868264 bytes dropped: 200921 packets 201724684 bytes
        Level 4: 25394 packets 25495576 bytes dropped: 251184 packets 252188736 bytes
```

1.4.7. Configuring Lawful Intercept (LI)



This section is still a work in progress.

1.5. Appendixes

Appendix A: RBFS Multiservice Edge Configuration

The RBFS Multiservice Edge configuration file (`multiservice.json`) can be downloaded from here.

- Click here to download the RBFS Multiservice Edge configuration file.

Appendix B: TACACS+ Server Configuration

The TACACS+ server configuration file (`tac_plus.conf`) can be downloaded from here.

- Click here to download the TACACS+ server configuration file.

Appendix C: RADIUS Server Configuration

The RADIUS server configuration files (`radius_config.zip`) can be downloaded from here. The zip archive contains the set of configuration files needed to configure the RADIUS server.

- Click here to download the RADIUS server configuration files.

Appendix D: BNG Blaster Configuration

The BNG Blaster configuration file (`blaster.json`) can be downloaded from here.

- Click here to download the BNG Blaster configuration files.

The JSON file (`isis_3node.json`) which is used to simulate R-1, R-2, and RR on BNG Blaster, can be downloaded from here.

- Click here to download the `isis_3node.json` file.

2. RBFS Multiservice Edge for IPoE Subscribers

2.1. An Overview of RBFS Multiservice Edge for IPoE Subscribers

This document provides guidance to validate IPoE implementation using the RBFS Multiservice Edge. The guide contains quick information about general platform configuration, configuration of various access and routing protocols, subscriber management, Quality of Service (QoS) and troubleshooting. The document presents a single use case scenario and provides information specifically on how to validate this particular implementation and for more information on any specific application, refer to https://documents.rtbrick.com/.

This guide is not intended to be an exhaustive guide of all RBFS features and does not provide information on features such as Multicast, Lawful Intercept etc.

2.1.1. About the RBFS Multiservice Edge

The RtBrick Multiservice Edge is delivered as a container, running on Rtbrick Host provided by the hardware ODM manufacturers. Platforms that support Multiservice Edge include Edgecore AGR 400, CSR 320, and UfiSpace S9600. The RtBrick Multiservice Edge software runs on powerful bare-metal switches as an open BNG.

The BNG is designed to dynamically deliver the following services:

1. Discovering and managing subscriber sessions for IPoE subscribers
2. Providing authentication, authorization and accounting (AAA)

The basic Multiservice Edge architecture for IPoE subscribers is shown in the figure below.

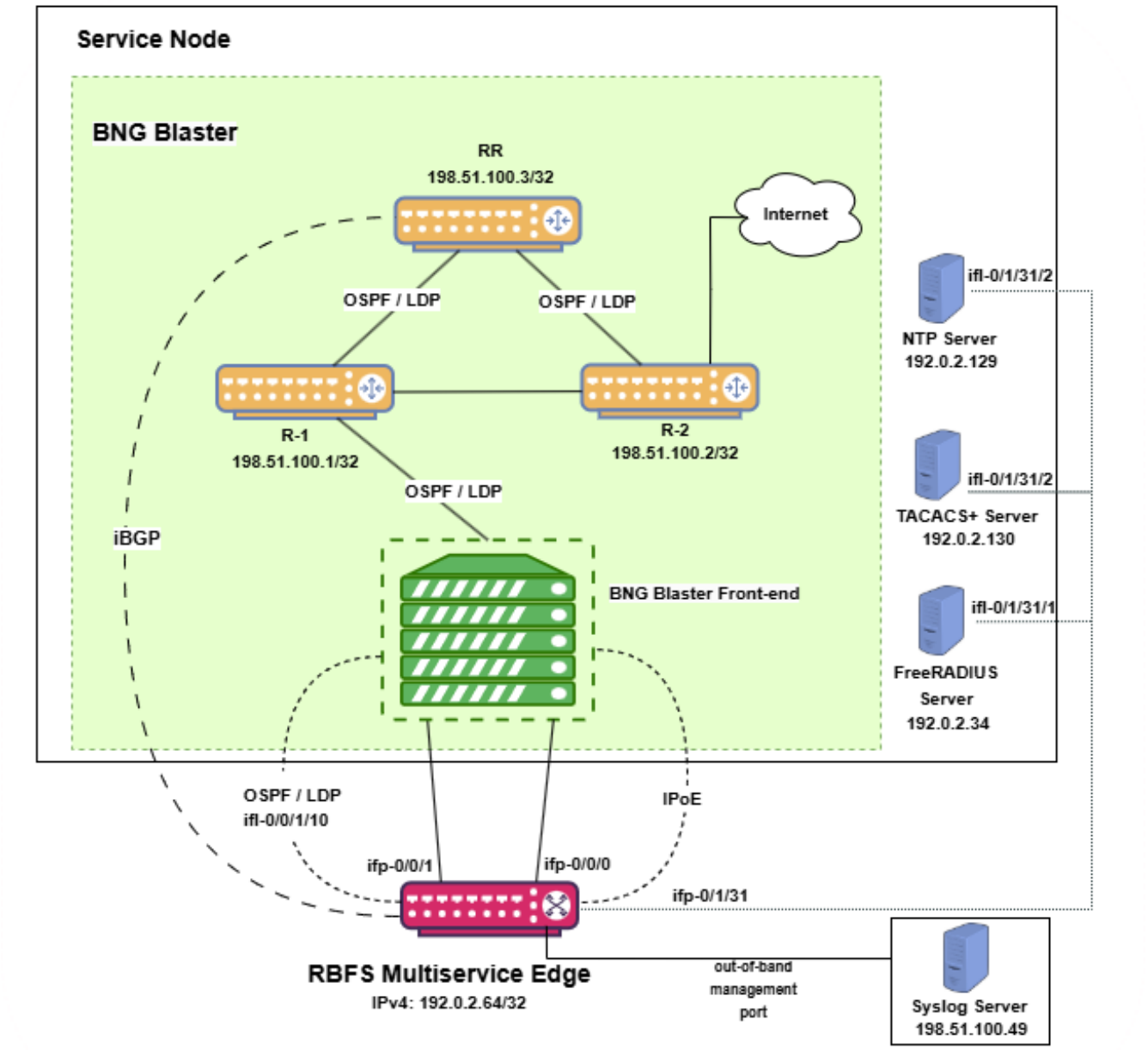


Figure 6. Multiservice Edge Architecture for IPoE Subscribers

Fig. 1: Topology setup with `Multiservice Edge` as a DUT (device under test) connected to BNG Blaster.

In this topology:

1. The Service Node is an Ubuntu server on which a new container is spawned with the associated interfaces for running BNG Blaster tests.
2. BNG Blaster emulates both the routing and access functions. It is an open-source network tester developed by RtBrick to validate the performance and scalability of BNG devices. The BNG blaster can emulate both subscribers at scale and network elements such as routers. Therefore, it is possible to create complex topologies with just the DUT and a server that hosts the BNG Blaster, thereby minimizing the equipment needed to test the DU. For more information, refer to the BNG Blaster documentation. In this topology, BNG Blaster emulates IPoE (DHCP) subscribers. The Service Node runs BNG Blaster. To use BNG Blaster, download and install the

BNG Blaster software onto the system that meets the tool's system requirements. The software can be downloaded from https://github.com/rtbrick/bngblaster/releases.

3. The topology emulates IPoE subscribers and traffic between RBFS switch and the core network.
4. The objective of this topology is to demonstrate complete IPoE subscriber emulating and service along with routing to connect to the network uplink.
5. R-1, R-2 and RR are simulated using BNG Blaster. The `Multiservice Edge` forms an OSPF L1 adjacency with the BNG blaster to discover R-1, R-2, and RR. The RR is the route reflector that replays a full internet feed of IPv4 and IPv6 addresses, in total about 1.1 million prefixes. R-2 is the nexthop for all internet routes.
6. Through DCHP, an IPv4 address needs to be assigned to the out-of-band management port.

2.1.2. Deployment

A Multiservice Edge provides BNG functionality on a single bare-metal switch and eliminates the need to have a chassis-based system. It provides a low footprint and optimal power consumption based on BRCM chipsets, a compelling value proposition that has complete BNG and routing feature support.

Multiservice Edge runs on small form-factor temperature-hardened hardware that allows deployments in street site cabinets.

For more information, see RBFS Installation Guide.

2.1.3. Using the RBFS CLI

Connect to the `Multiservice Edge` node.

```
$ ssh <multiservice-edge-management-ip> -l supervisor
supervisor@<multiservice-edge-management-ip>'s password:
```

The password for `multiservice-edge-management-ip` should be entered here.

As a result, the CLI prompt will look like this:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net:~ $
```

Open the RBFS CLI.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net:~ $ cli
```

The CLI has three different modes:

- `operation` mode is a read-only mode to inspect and analyse the system state
- `config` mode allows modifying the RBFS configuration

- `debug` mode provides advanced tools for trouble-shooting

The `switch-mode` command allows switching between the different modes. The `show` commands allow inspecting the system state. The `set` and `delete` commands, which are only available in the configuration mode, allow modifying or deleting the current configuration. The `commit` command persists the changes. RBFS provides a commit history which allows reviewing changes (`show commit log`) and restoring a previous configuration (`rollback`). There are also commands to `ping` destinations, `capture` network traffic, `save` the configuration to or `load` the configuration from a file.

The CLI supports abbreviating commands, provides suggestions by hitting the `[tab]` key and displays a context help by entering a `?`.

For more information on how to use the RBFS CLI, see the RBFS CLI User Guide.

2.2. Configuration and Settings

2.2.1. Platform Configuration and Settings

This section provides information about the platform and how to set various required configurations for the platform.

Know your Device

The configurations provided in this reference design document (Multiservice Edge IPoE implementation) are generated on the UfiSpace S9600-72XC platform. 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 [1GbE/10GbE/25GbE] and 8x100GE [40GbE/100GbE] high-speed ports with a switching capacity of up to 2.4Tbs.

The RBFS Multiservice Edge software is installed on top of the UfiSpace S9600-72XC.



Although the specific device used here is UfiSpace S9600-72XC, the configuration will stay exactly the same for any other device that supports the Multiservice Edge image.

For more information about the hardware specifications of UfiSpace S9600-72XC, see Platform Guide.

Prerequisites

- Access to BNG Blaster, an open-source network testing platform for access and routing protocols. For information on obtaining and building BNG Blaster, see https://rtbrick.github.io/bngblaster/.
- Access to FreeRADIUS, a free RADIUS suite. For accessing FreeRADIUS, see https://freeradius.org/.
- Access to Syslog server.

Restore Configuration

Depending on the deployment scenario, a running configuration can be applied or restored as needed.

To enable configuration restore, enter the `set system load-last-config true` command as shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> set system load-last-config true
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> commit
```

For more information, see the section "Running Configuration" of the RBFS NOC Troubleshooting Guide.

2.2.2. General Configuration

To enable testing some basic primitives need to be configured. These general configurations include loopback interface for identifying and accessing the device on network, NTP for setting accurate time across a whole network of devices, TACACS+ for user authentication, user management for user configuration, license for accessing RBFS, Resmon for resource monitoring, and Syslog configurations for exporting the log message to the external log management server.

Configure License

Without any license installed on your system, you can evaluate RBFS for 7 days. You need to get an evaluation license or purchase an actual license within 7 days to use the full functionality of RBFS.

The following steps provide the commands to install an RBFS license key. For more information about license configuration, see Installing License.

Switch to config mode using the switch-mode config command to continue with the RBFS configurations.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> switch-mode config
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg>
```

Install the license encrypted string (that is received from RtBrick) using the RBFS CLI.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> set system license <license-key>
```

RBFS license configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show config system license
AAAAWsfg&jdkfs4D34H5@2evf...
```

As shown below, the "show system license" command displays the expiration date for the current

license.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show system license
License Validity:
  License index 1:
    Start date : Tue Feb 28 09:44:27 GMT +0000 2023
    End date   : Mon Mar 04 09:44:27 GMT +0000 2024
supervisor@rtbrick>multiservice-edge.rtbrick.net: op>
```

Configure Instance

Instance `default` will be available by default without any configurations.

Create the instance `inband_mgmt` by entering the following commands.

```
set instance inband_mgmt
set instance inband_mgmt address-family ipv4 unicast
commit
```

The configurations of the instance `inband_mgmt` are shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config instance inband_mgmt
{
  "rtbrick-config:instance": [
    {
      "name": "inband_mgmt",
      "address-family": [
        {
          "afi": "ipv4",
          "safi": "unicast"
        }
      ]
    }
  ]
}
```

Below are the configurations available for the available instances.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show instance detail
Instance: default
  Instance ID: 0
  State: Active
  AFI        SAFI             State
  ipv4       unicast          Active
  ipv4       multicast        Active
  ipv4       labeled-unicast  Active
  ipv6       unicast          Active
```

```
  ipv6       multicast        Active
  ipv6       labeled-unicast  Active
  mpls       unicast          Active
Instance: inband_mgmt
  Instance ID: 3
  State: Active
  AFI        SAFI             State
  ipv4       unicast          Active
```

Configure Loopback Interface

Loopback Interface configuration is required as it is the best way to identify a network device and is always reachable. Also, protocols use the loopback address to determine protocol-specific properties for the device.

The following steps provide the commands to configure the loopback interface. For more information about Loopback Interface configuration, see Interfaces User Guide.

Configure loopback interface on the device.

```
set interface lo-0/0/1 unit 0 address ipv4 192.0.2.64/32
set interface lo-0/0/1 unit 1 address ipv4 192.0.2.74/32
set interface lo-0/0/1 unit 2 instance inband_mgmt
set interface lo-0/0/1 unit 2 address ipv4 192.0.2.128/32
set interface lo-0/0/1 unit 3 instance inband_mgmt
set interface lo-0/0/1 unit 3 address ipv4 192.0.2.131/32
commit
```



The configuration commands should be followed by the `commit` command to save the configurations into the device.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> commit
```

Loopback Interface configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config interface lo-0/0/1
{
  "rtbrick-config:interface": [
    {
      "name": "lo-0/0/1",
      "unit": [
        {
          "unit-id": 0,
          "address": {
            "ipv4": [
              {
                "prefix4": "192.0.2.64/32"
```

```
            }
          ]
        }
      },
      {
        "unit-id": 1,
        "address": {
          "ipv4": [
            {
              "prefix4": "192.0.2.74/32"
            }
          ]
        }
      },
      {
        "unit-id": 2,
        "instance": "inband_mgmt",
        "address": {
          "ipv4": [
            {
              "prefix4": "192.0.2.128/32"
            }
          ]
        }
      },
      {
        "unit-id": 3,
        "instance": "inband_mgmt",
        "address": {
          "ipv4": [
            {
              "prefix4": "192.0.2.131/32"
            }
          ]
        }
      }
    ]
  }
 ]
}
```

Configure IP Addresses for Core Interfaces

Enter the following commands to configure IP addresses for the core interfaces.

```
set interface ifp-0/0/1 unit 10
set interface ifp-0/0/1 unit 10 vlan 10
set interface ifp-0/0/1 unit 10 address ipv4 192.0.2.1/27
set interface ifp-0/0/1 unit 10 address ipv6 2001:db8::1/64
set interface ifp-0/0/1 unit 100
```

```
set interface ifp-0/0/1 unit 100 vlan 100
set interface ifp-0/0/1 unit 100 address ipv4 192.0.2.33/27
set interface ifp-0/0/1 unit 200
set interface ifp-0/0/1 unit 200 instance inband_mgmt
set interface ifp-0/0/1 unit 200 vlan 200
set interface ifp-0/0/1 unit 200 address ipv4 192.0.2.97/27
commit
```

Below configuration shows the IP address configurations for the core interfaces.

```
{
    "rtbrick-config:interface": [
      {
        "name": "ifp-0/0/1",
        "unit": [
          {
            "unit-id": 10,
            "vlan": 10,
            "address": {
              "ipv4": [
                {
                  "prefix4": "192.0.2.1/27"
                }
              ],
              "ipv6": [
                {
                  "prefix6": "2001:db8::1/64"
                }
              ]
            }
          },
          {
            "unit-id": 100,
            "vlan": 100,
            "address": {
              "ipv4": [
                {
                  "prefix4": "192.0.2.33/27"
                }
              ]
            }
          },
          {
            "unit-id": 200,
            "instance": "inband_mgmt",
            "vlan": 200,
            "address": {
              "ipv4": [
                {
                  "prefix4": "192.0.2.97/27"
```

```
                }
              ]
            }
          }
        ]
      }
    ]
  }
}
```

Configure Static Routes to Enable Reachability to the NTP and TACACS Servers

Below are static routes configured to enable reachability to the NTP (192.0.2.129) and TACACS (192.0.2.130) servers. On the Service Node, 192.0.2.98 is the interface address on VLAN 200. It is explained later in this document how to configure IP addresses on Service Node. For details, see Configuring Interfaces on the Service Node for NTP and TACACS Connectivity from Multiservice Edge.

```
set instance inband_mgmt static route ipv4 192.0.2.129/32 unicast np1
set instance inband_mgmt static route ipv4 192.0.2.130/32 unicast np1
set instance inband_mgmt static nexthop-profile np1
set instance inband_mgmt static nexthop-profile np1 nexthop 192.0.2.98
set instance inband_mgmt static nexthop-profile np1 lookup-instance inband_mgmt
set instance inband_mgmt static nexthop-profile np1 lookup-afi ipv4
set instance inband_mgmt static nexthop-profile np1 lookup-safi unicast
commit
```

The configuration of the static routes is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config instance inband_mgmt static
{
  "rtbrick-config:static": {
    "route": {
      "ipv4": [
        {
          "prefix4": "192.0.2.129/32",
          "safi": "unicast",
          "nexthop-profile": "np1"
        },
        {
          "prefix4": "192.0.2.130/32",
          "safi": "unicast",
          "nexthop-profile": "np1"
        }
      ]
    },
    "nexthop-profile": [
      {
        "name": "np1",
        "nexthop": "192.0.2.98",
```

```
        "lookup-instance": "inband_mgmt",
        "lookup-afi": "ipv4",
        "lookup-safi": "unicast"
      }
    ]
  }
}
```

Configure NTP

Configuring NTP (Network Time Protocol) provides time synchronization across a whole network of devices. An NTP network consists of devices (clients) that are to be synchronized with the NTP server that provides accurate time to the client devices.

The following steps provide the commands to configure Network Time Protocol (NTP) for the device. For more information about NTP configuration, see NTP User Guide.

Enabling NTP Service:

To access the NTP service running in the RtBrick Host, this service has to be enabled in inband-management. On configuring this, the hosts reachable in inband instance via the physical interface can access this service.

Configure NTP server and NTP service on the device.

```
set system ntp server ntp1
set system ntp server ntp1 ipv4-address 192.0.2.129
set inband-management instance inband_mgmt
set inband-management instance inband_mgmt ntp true
commit
```

NTP configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config inband-management
{
    "rtbrick-config:inband-management": {
      "instance": [
        {
          "name": "inband_mgmt",
          "ntp": "true"
        }
      ]
    }
  }
```

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config system ntp
{
    "rtbrick-config:ntp": {
```

```
      "server": [
        {
          "server-name": "ntp1",
          "ipv4-address": "192.0.2.129"
        }
      ]
    }
  }
```

User Authentication

RBFS supports user authentication through a centralized TACACS+ server and with a local authentication system. The following authentication process typically occurs when a user attempts to access the network.

1. When a user logs in through SSH, the SSH Daemon (sshd) invokes the Pluggable Authentication Module (PAM) to trigger authentication process.
2. PAM requests TACACS+ authentication (except for the user with the supervisor privileges).
3. TACACS+ server provides 'grant access' node if the user authentication is successful.
4. If the user is not allowed using the TACACS+ authentication, it is required to undergo an additional authentication phase. PAM looks up local users. Upon successful authentication, PAM generates RTB PAM token; includes user role in 'scope'.

Define Users on TACACS+ Server

Administrator needs to define users and associate them with the predefined roles on the TACACS+ server. Optionally, RBFS CLI commands can be restricted using the rtb-allow-cmds and rtb-deny-cmds.

The tac_plus.conf file contains configuration information for the tac_plus(tacacs+) daemon. This file is stored at the following location:

/etc/tacacs+/tac_plus.conf

To view the TACACS+ server configuration file, enter the following command.

```
sudo cat /etc/tacacs+/tac_plus.conf
```

For more information about TACACS+ server configuration, see

https://manpages.ubuntu.com/manpages/trusty/man5/tac_plus.conf.5.html

This Reference Design document uses the default local user supervisor for the configurations, whereas other users, defined in the TACACS server, can log into RBFS by using their usernames and passwords.

The following TACACS+ configuration shows the details of the TACACS users.

- Click here to download the `tac_plus.conf` ile.

Configure TACACS+ on RBFS

After defining the users on the TACACS+ server, configure the TACACS+ server on Multiservice Edge. This configuration allows the remote TACACS+ server to communicate with the Multiservice Edge and to validate user access on the network.

The following steps provide the commands to configure TACACS+. For more information about TACACS+ configuration, see Configure TACACS+ on RBFS.

To access the TACACS+ service running in the RtBrick Host, this service has to be enabled in inband management. On configuring this, the hosts reachable in inband instance via the physical interface can access this service.

```
set system secure-management-status true
set system authorization tacacs 192.0.2.130 inband secret-plain-text RtBrick_Little_Secret
set inband-management instance inband_mgmt tacacs true
commit
```

In the above configuration, the command `set inband-management instance inband_mgmt tacacs true` is used to enable TACACS+ under the instance called `inband_mgmt`.

TACACS+ configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config system authorization
{
  "rtbrick-config:authorization": {
    "tacacs": [
      {
        "ipv4-address": "192.0.2.130",
        "type": "inband",
        "secret-encrypted-text": "$22464b2c7336cfe71e596c447be28d598b9b7b37f92faea157fd5058e5fe0d769"
      }
    ]
  }
}
```

Configuration for enabling TACACS+ under the instance `inband_mgmt` is shown below:

```
"rtbrick-config:inband-management": {
      "instance": [
        {
          "name": "inband_mgmt",
          "tacacs": "true"
        }
      ]
    },
```

Enabling TACACS+ Service on the Service Node

Enter the following commands to enable the TACACS service on the Service Node.

```
~$ sudo /bin/systemctl enable tacacs_plus.service
~$ sudo /bin/systemctl start tacacs_plus.service
```

Validating TACACS+ authentication

The following scenario shows a successful authentication for the user `bob` with password `bob`.

```
~$ ssh bob@multiservice-edge.rtbrick.net
bob@multiservice-edge.rtbrick.net's password:
Last login: Mon Apr 3 16:13:40 2023 from multiservice-edge.rtbrick.net
bob@rtbrick>multiservice-edge.rtbrick.net: op>
```

The following scenario shows an unsuccessful password authentication for the user `bob` with password `bob123`.

```
~$ ssh bob@rtbrick>multiservice-edge.rtbrick.net:
bob@multiservice-edge.rtbrick.net's password:
Permission denied, please try again.
bob@multiservice-edge.rtbrick.net's password:
```

The following scenario shows an unsuccessful authentication for an undefined user `frank`.

```
~$ ssh frank@rtbrick>multiservice-edge.rtbrick.net:
frank@multiservice-edge.rtbrick.net's password:
Permission denied, please try again.
frank@multiservice-edge.rtbrick.net's password:
accounting file = /var/log/tac_plus.acct
key = RtBrick_Little_Secret
```

Configure User Management

Configuring Local User Management enables administrators to create, manage, and secure the users and groups. It allows creation of privileges that are configurable for user-defined and predefined roles.

The following steps provide the commands to configure user management. For more information, see Local User Management.

1. To create a role, configure the RBAC privilege and the command privilege. To configure the RBAC privilege for both table and object:

```
set system user admin role supervisor
```

```
set system user admin shell /bin/bash
set system user admin password-hashed-text
$6$XNkmuMRI.5.R/NBJ$XDfZec7gEM3z/3lYn8mDDWimRZ/68xawia.pTMdrGqoYHEE3nWHB08DeaPNQTwHW6WjB1aX6.xjYjh8CNCy4g1
commit
```

For information about Configuring hashed passwords, see Configure Hashed Password.

The authentication configuration of a password hashed text and an SSH public key is shown below:

```
{
  "ietf-restconf:data": {
    "rtbrick-config:system": {
      "user": [
        {
          "username": "admin",
          "shell": "/usr/local/bin/cli",
          "password-hashed-text": "$5$L2DaOYYuddhBV$9RA5MX9RQzLC9fIKJzbnoFBb88w9rkSXl7GVrVJ9PY7",
          "ssh-pub-key": [
            "ssh-rsa AAAAWsfg&jdkfs4D34H5@2evf....."
            ]
        }
      ]
    }
  }
}
```

Configure Syslog

RBFS supports sending log messages to a Syslog server. The Syslog configuration can be performed in RBFS.

To configure logging for bgp by using Syslog, enter the following commands.

```
set log module bgp
set log module bgp level debug
set log module bgp plugin-alias
set log module bgp plugin-alias alias-name syslog
set log module bgp plugin-alias level debug
commit
```



For event logging, CtrlD only supports Graylog and Syslog. Graylog must be disabled in order to enable Syslog. In addition, Graylog attributes must be replaced with Syslog attributes.

Sylog configuration for the module bgp is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config log
{
  "rtbrick-config:log": {
```

```
    "module": [
      {
        "module-name": "bgp",
        "level": "debug",
        "plugin-alias": {
          "alias-name": "syslog",
          "level": "debug"
        }
      }
    ]
  }
}
```

Accessing the RtBrick Host to Configure Syslog

The steps described in this section are performed on the RtBrick Host. For logging into the RtBrick Host, use SSH port 1022.

```
ssh supervisor@<multiservice-edge-management-ip> -p 1022
```

After logging into the RtBrick Host, go to the following location of CtrlD and edit the `config.json` file.

- `/etc/rtbrick/ctrld/config.json`

Specify the Syslog configurations as shown below in the `config.json` file.

```
{
    "rbms_enable": false,
    "graylog_enable": false,
    "syslog_enable": true,
    "syslog_network": "udp",
    "syslog_urls": [
      "198.51.100.49:516"
    ],
    "syslog_severity_level": 7,
    "auth_enabled": false
}
```



- For documentation purposes, the IP address `198.51.100.49` has been used as the IP address of the Syslog endpoint. This IP address should be updated with the actual Syslog server's IP address.
- Syslog messages can be transported using UDP or TCP protocol. In this configuration, Syslog messages are transported using `udp`.

After making configuration changes in the `config.json`, restart CtrlD service as shown below.

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

```
[sudo] password for supervisor:
[ ok ] Stopping rtbrick ctrld service:.
[ ok ] Starting rtbrick ctrld service:.
```

Monitor Resources (Resmon)

Resource monitoring enables administrators to collect and analyze the health information and usage data of various hardware resources such as CPU, memory, processes, disks, sensors, optics, and so on.

Run `show cpu usage`, `show memory usage` and `show disk usage` to see the CPU, memory and disk utilization respectively.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show cpu usage
Name         Total       User     System       Nice   I/O Wait       Idle        IRQ   Soft IRQ
cpu            11%         9%         1%         0%         0%        88%         0%         0%
cpu0           10%         8%         1%         0%         0%        89%         0%         0%
cpu1           44%        43%         1%         0%         0%        55%         0%         0%
cpu2           34%        32%         2%         0%         0%        66%         0%         0%
cpu3           11%         9%         2%         0%         0%        89%         0%         0%
cpu4            3%         1%         2%         0%         0%        96%         0%         0%
cpu5            4%         3%         1%         0%         0%        96%         0%         0%
cpu6           13%        11%         2%         0%         0%        87%         0%         0%
cpu7           24%        22%         1%         0%         0%        75%         0%         0%
cpu8            4%         2%         2%         0%         0%        95%         0%         0%
cpu9            2%         1%         1%         0%         0%        97%         0%         0%
cpu10           6%         4%         2%         0%         0%        93%         0%         0%
cpu11           3%         2%         1%         0%         0%        96%         0%         0%
cpu12           8%         6%         2%         0%         0%        91%         0%         0%
cpu13           3%         3%         0%         0%         0%        96%         0%         0%
cpu14           6%         4%         2%         0%         0%        93%         0%         0%
cpu15           8%         5%         2%         0%         0%        91%         0%         0%
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show memory usage
Name     Total            Used             Free          Shared       Buffers      Cached
RAM      31.03 GiB        8.51 GiB         17.04 GiB     1.19 GiB     112.66 MiB   5.37 GiB
SWAP     0 bytes          0 bytes          0 bytes       n/a          n/a          n/a
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show disk usage
Filesystem               Type             Size         Used         Available    Mountpoint          Usage %
none                     tmpfs            492 KiB      0 bytes      492 KiB      /dev                0.0
tmpfs                    tmpfs            15.51 GiB    17.23 MiB    15.5 GiB     /run                0.11
tmpfs                    tmpfs            6 GiB        828.75 MiB   5.19 GiB     /shm                13.49
tmpfs                    tmpfs            15.51 GiB    182.96 MiB   15.33 GiB    /dev/shm            1.15
tmpfs                    tmpfs            5 MiB        0 bytes      5 MiB        /run/lock           0.0
devtmpfs                 devtmpfs         1 MiB        0 bytes      1 MiB        /dev/mem            0.0
/dev/sda10               ext4             15.62 GiB    50.61 MiB    14.76 GiB    /var/log            0.33
/dev/sda6                ext4             29.4 GiB     4.24 GiB     23.65 GiB    /platform           15.2
tmpfs                    tmpfs            3.1 GiB      0 bytes      3.1 GiB      /run/user/1000      0.0
tmpfs                    tmpfs            3.1 GiB      0 bytes      3.1 GiB      /run/user/1001      0.0
tmpfs                    tmpfs            15.51 GiB    0 bytes      15.51 GiB    /sys/fs/cgroup      0.0
/dev/sda11               ext4             43.79 GiB    51.89 MiB    41.49 GiB    /var/crash          0.12
tmpfs                    tmpfs            3.1 GiB      1.02 MiB     3.1 GiB      /var/run-ext/onl/r  0.03
/var/cache/rtbrick/imag  overlay          29.4 GiB     4.24 GiB     23.65 GiB    /                   15.2
```

The show command can also be used to view other resource details. For information about the resmon configuration and operational commands, see the RBFS Resource Monitoring Guide.

2.3. Protocol Configurations

This validated solution design topology uses OSPF as the interior gateway protocol to distribute IP routing information among the routers in an Autonomous System (AS). The Label Distribution Protocol (LDP) is used to exchange label mapping information for MPLS traffic. And, iBGP is used for exchanging routing and reachability information within ASs.

One thus needs to configure the following protocols:

- OSPF : To ensure IP connectivity on the core network.
- LDP : To establish MPLS LSP tunnels for MPLS data transmission on the network.
- iBGP: To exchange routing information within an AS.

2.3.1. Configure OSPF

The following steps provide the commands to execute various functionalities of the OSPF protocol. For more detailed information about OSPF configuration, see OSPF User Guide.

1. Configure OSPF address-family, hostname, area-id, and interfaces.

```
set instance default protocol ospf address-family ipv4 hostname Multiservice-Edge.
set instance default protocol ospf address-family ipv4 segment-routing srgb base 100000
set instance default protocol ospf address-family ipv4 segment-routing srgb range 2000
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-0/0/1/10
set instance default protocol ospf address-family ipv4 area 0.0.0.0 interface ifl-0/0/1/10 network-type p2p
set instance default protocol ospf address-family ipv4 area 0.0.0.0 interface lo-0/0/1/0
set instance default protocol ospf address-family ipv4 area 0.0.0.0 interface lo-0/0/1/0 segment-routing index 100
commit
```

OSPF instance configuration on the interface is shown below:

```
supervisor@rtbrick.net: op> show config instance default protocol ospf
{
  "rtbrick-config:ospf": {
    "address-family": {
      "ipv4": {
        "hostname": "Multiservice-Edge.",
        "segment-routing": {
          "srgb": {
            "base": 100000,
            "range": 2000
          }
        },
        "area": [
          {
            "area-id": "0.0.0.0",
            "interface": [
              {
```

```
                "name": "ifl-0/0/1/10",
                "network-type": "p2p"
              },
              {
                "name": "lo-0/0/1/0",
                "segment-routing": {
                  "index": 100
                }
              }
            ]
          }
        ]
      }
    }
  }
}
```

2.3.2. Configure LDP on the Interfaces

The following steps provide the commands to execute various LDP functionalities. For more detailed information about LDP configuration, see *LDP User Guide*.

1. Configure LDP on the router interface.

```
set instance default protocol ldp router-id 192.0.2.64
set instance default protocol ldp interface ifl-0/0/1/10
set instance default protocol ldp interface lo-0/0/1/0
commit
```

Configuration for LDP on the interface is shown below:

```
supervisor@rtbrick.net: cfg> show config instance default protocol ldp
{
  "rtbrick-config:ldp": {
    "router-id": "192.0.2.64",
    "interface": [
      {
        "name": "ifl-0/0/1/10"
      },
      {
        "name": "lo-0/0/1/0"
      }
    ]
  }
}
```

2.3.3. Configure BGP

The following steps provide the commands to execute the various BGP functionalities quickly. For more detailed information about BGP configuration, see BGP User Guide.

1. Configure BGP local AS, router-id, and hostname

```
set instance default protocol bgp local-as 4200000001
set instance default protocol bgp router-id 192.0.2.64
commit
```

BGP local AS, router-id, and hostname configurations are shown below:

```
supervisor@rtbrick.net: cfg> show config instance default protocol bgp
{
  "rtbrick-config:bgp": {
    "local-as": 4200000001,
    "router-id": "192.0.2.64",

<....>
```

2. Enable the IPv4 and IPv6 address families which are to be supported on the specific BGP instance.

```
set instance default protocol bgp address-family ipv4 unicast
set instance default protocol bgp address-family ipv4 unicast resolve-nexthop safi labeled-unicast
set instance default protocol bgp address-family ipv4 unicast redistribute direct
set instance default protocol bgp address-family ipv6 labeled-unicast
set instance default protocol bgp address-family ipv6 unicast
set instance default protocol bgp address-family ipv6 unicast resolve-nexthop safi labeled-unicast
set instance default protocol bgp address-family ipv6 unicast redistribute direct
commit
```

BGP address family configuration is shown below:

```
supervisor@rtbrick.net: cfg> show config instance default protocol bgp address-family
{
  "rtbrick-config:address-family": [
    {
      "afi": "ipv4",
      "safi": "unicast",
      "resolve-nexthop": {
        "safi": "labeled-unicast"
      },
      "redistribute": [
        {
          "source": "direct"
        }
```

```
      ]
    },
    {
      "afi": "ipv6",
      "safi": "labeled-unicast"
    },
    {
      "afi": "ipv6",
      "safi": "unicast",
      "resolve-nexthop": {
        "safi": "labeled-unicast"
      },
      "redistribute": [
        {
          "source": "direct"
        }
      ]
    }
  ]
}
```

3. Create the peer group with the specific remote AS configurations and the address family that is to be negotiated with the peer which will be attached to the peer group later.

```
set instance default protocol bgp peer-group RR
set instance default protocol bgp peer-group RR remote-as 4200000001
set instance default protocol bgp peer-group RR address-family ipv4 unicast
set instance default protocol bgp peer-group RR address-family ipv6 labeled-unicast
set instance default protocol bgp peer-group RR address-family ipv6 unicast
set instance default protocol bgp peer-group RR address-family ipv6 unicast send-label true
commit
```

BGP peer-group configuration is shown below:

```
supervisor@rtbrick.net: cfg> show config instance default protocol bgp peer-group
{
  "rtbrick-config:peer-group": [
    {
      "pg-name": "RR",
      "remote-as": 4200000001,
      "address-family": [
        {
          "afi": "ipv4",
          "safi": "unicast"
        },
        {
          "afi": "ipv6",
          "safi": "labeled-unicast"
        },
```

```
        {
          "afi": "ipv6",
          "safi": "unicast",
          "send-label": "true"
        }
      ]
    }
  ]
}
```

4. Add a BGP peer and associate it with the specific peer group.

```
set instance default protocol bgp peer
set instance default protocol bgp peer ipv4 198.51.100.3 192.0.2.64
set instance default protocol bgp peer ipv4 198.51.100.3 192.0.2.64 peer-group RR
commit
```

Configuration for adding a BGP peer and associating it with a peer group is shown below:

```
supervisor@rtbrick.net: cfg> show config instance default protocol bgp peer
{
  "rtbrick-config:peer": {
    "ipv4": [
      {
        "peer-address": "198.51.100.3",
        "update-source": "192.0.2.64",
        "peer-group": "RR"
      }
    ]
  }
}
```

5. Configure the IPv6 unicast address family with `send-label` as true, then address-family IPv6 labeled-unicast gets negotiated with the peer.

```
set instance default protocol bgp peer-group RR address-family ipv6 unicast send-label true
commit
```

The following configuration shows the BGP IPv6 unicast address family with `send-label` as true.

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

```
  ]
}
```

6. Set the resolve-nexthop, if the BGP nexthop attribute of the BGP routes needs to be resolved under ipv4/ipv6 labeled-unicast routing table. It configures only resolve-nexthop safi. Based on the nexthop-type (ipv4 or ipv6), it gets looked up into either IPv4 labeled-unicast or IPv6 labeled-unicast.

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

Resolve nexthop configuration is shown below:

```
supervisor@rtbrick.net: cfg> show config instance default protocol bgp address-family ipv4 unicast resolve-nexthop
{
  "rtbrick-config:resolve-nexthop": {
    "safi": "labeled-unicast"
  }
}
```

7. To redistribute the routes (belonging to a specific source) into BGP, execute the following command. The following command redistributes `direct` routes into BGP.

```
set instance default protocol bgp address-family ipv4 unicast redistribute direct
commit
```

```
supervisor@rtbrick.net: cfg> show config instance default protocol bgp address-family ipv4 unicast redistribute
{
  "rtbrick-config:redistribute": [
    {
      "source": "direct"
    }
  ]
}
```

2.3.4. Configuring the Service Node

Configuring Interfaces on the Service Node for RADIUS Connectivity from Multiservice Edge

To configure interfaces on the Service Node for RADIUS connectivity from Multiservice Edge, enter the following commands:

```
sudo ip link add link SN1-3-C1 name SN1-3-C1.100 type vlan id 100
sudo ifconfig SN1-3-C1.100 192.0.2.34/27
```



`SN1-3-C1` is the internal nomenclature that denotes the interface name on Service

> Node that connects to the Multiservice Edge.

Configuring Interfaces on the Service Node for NTP and TACACS Connectivity from Multiservice Edge

To configure interfaces on the Service Node for NTP and TACACS connectivity from Multiservice Edge, enter the following commands:

```
sudo ip link add link SN1-3-C1 name SN1-3-C1.200 type vlan id 200
sudo ifconfig SN1-3-C1.200 192.0.2.98/27
sudo ifconfig lo:1 192.0.2.129 netmask 255.255.255.255 up
sudo ifconfig lo:2 192.0.2.130 netmask 255.255.255.255 up
```

Configuring Routes on the Service Node

To configure routes on the Service Node that provide reachability to the RADIUS, TACACS and NTP servers, enter the following commands:

```
sudo ip route add 192.0.2.74/32 via 192.0.2.33
sudo ip route add 192.0.2.131/32 via 192.0.2.97
sudo ip route add 192.0.2.128/32 via 192.0.2.97
```

BNG Blaster Configuration for Protocols

BNG Blaster is an open-source network testing platform for access and routing protocols. It can emulate massive PPPoE and IPoE (DHCP) subscribers including IPTV, and L2TP (LNS). There are various routing protocols supported such as OSPF and BGP. So, one can use this platform for end-to-end BNG and non-BNG router testing.

For more information about BNG Blaster, see https://github.com/rtbrick/bngblaster

For information about installing BNG Blaster, see https://rtbrick.github.io/bngblaster/install.html

Downloading the Blaster Configuration File

The following is the configuration file that is used in BNG Blaster for validating IPoE, BGP, OSPF, and LDP.

- Click here to download the BNG Blaster configuration file (`blaster.json`).

Generating Supporting Files for Protocols

- **Generating BGP Internet Prefixes**

Enter the following commands to generate BGP internet prefixes.

```
bgpupdate -f internet.bgp -a 4200000001 -n 198.51.100.2 -N 1 -p 172.16.0.0/24 -P 1000000
bgpupdate -f internet.bgp -a 4200000001 -n 198.51.100.2 -N 1 -p 2004::/48 -m 10000 -M 5 -P 150000 --append
```

Ensure that the command execution has finished (as shown below) before continuing.

```
[2023-04-05 10:19:32][INFO   ] init 1000000 IPv4 prefixes
[2023-04-05 10:19:56][INFO   ] open file internet.bgp (replace)
[2023-04-05 10:25:31][INFO   ] finished
```

After the generation of the `internet.bgp` file, the "raw-update-file" attribute of the `blaster.json` file needs to be updated as follows:

```
"raw-update-file": "/home/supervisor/internet.bgp"
```



For more information about downloading the blaster.json file, see section Downloading the Blaster Configuration File.

- **Generating MRT FIle for OSPF**

Below is the JSON file (`ospf_3node.json`) which is used to simulate R-1, R-2, and RR on BNG Blaster.

- Click here to download the `ospf_3node.json` file.

This JSON file needs to be converted to MRT format using the following command:

```
lspgen -r ospf_3node.json -m ospf.mrt
```

After converting the file to `ospf.mrt`, it needs to be updated in the ospf section in the `blaster.json` file as shown below.

```
"mrt-file": "/home/supervisor/ospf.mrt"
```

- Generate labels for the OSPF prefixes using "ldpupdate" command as shown below:

```
ldpupdate -l 192.0.2.65 -p 198.51.100.1/32 -P 3 -M 3 -f out.ldp
```

The detail of the generated file needs to be added to the LDP section in the `blaster.json` file as shown below:

```
"raw-update-file": "/home/supervisor/out.ldp"
```

Starting BNG Blaster

In the following command line string, a BNG Blaster instance is started and the `blaster.json` file is used.

```
sudo bngblaster -C blaster.json -I
```

The **-C blaster.json** argument specifies the blaster configuration file. The -I flag enables the interactive blaster UI.

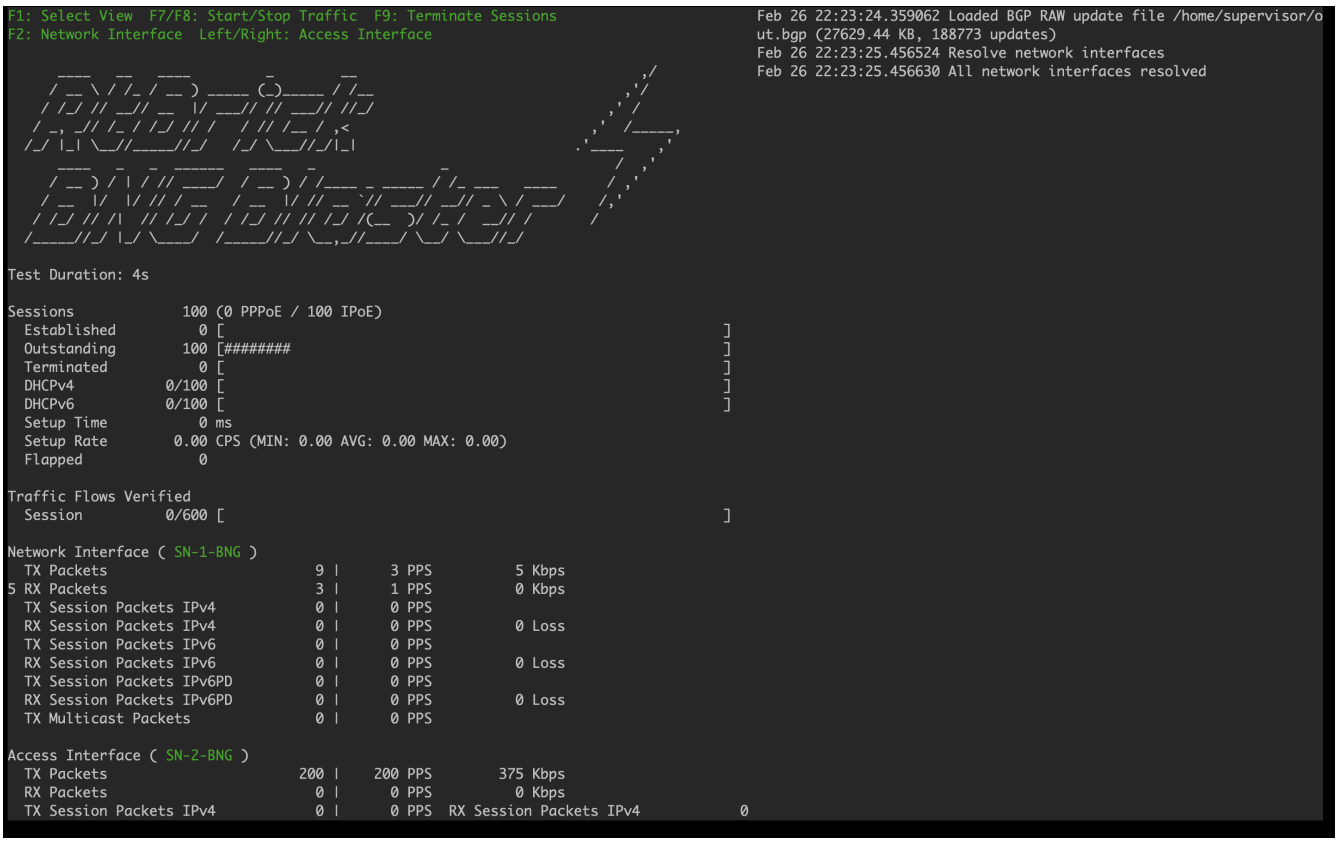


Figure 7. Starting BNG Blaster

2.3.5. Validating Protocols on RBFS Multiservice Edge

Validating OSPF Adjacency, Routes and Reachability

Run the following command to show OSPF adjacency.

```
supervisor@rtbrick.net: op> show ospf neighbor
Instance: default, Address family: ipv4
  Address                       Interface           Router          Area            State      Priority  DR
   BDR             Uptime          Expires
  192.0.2.2                     ifl-0/0/1/10      R1              0.0.0.0         Full       64        0.0.0.0
0.0.0.0         0d:00h:00m:03s  38s
```

After configuring OSPF protocol, check the IPv4 unicast routes, populated by OSPF using the following command:

```
supervisor@rtbrick.net: op> show route ipv4 unicast source ospf instance default
Instance: default, AFI: ipv4, SAFI: unicast
Prefix/Label                              Source            Pref    Next Hop                                  Interface
192.0.2.65/32                             ospf              10      192.0.2.2                                 ifl-0/0/1/10
```

```
198.51.100.1/32                          ospf            10      192.0.2.2                               ifl-
0/0/1/10
198.51.100.2/32                          ospf            10      192.0.2.2                               ifl-
0/0/1/10
198.51.100.3/32                          ospf            10      192.0.2.2                               ifl-
0/0/1/10
198.51.100.100/31                        ospf            10      192.0.2.2                               ifl-
0/0/1/10
198.51.100.102/31                        ospf            10      192.0.2.2                               ifl-
0/0/1/10
198.51.100.104/31                        ospf            10      192.0.2.2                               ifl-
0/0/1/10
```

Ping the address 192.0.2.65 as follows:

```
supervisor@rtbrick.net: op> ping 192.0.2.65
68 bytes from 192.0.2.65: icmp_seq=1 ttl=64 time=8.2474 ms
68 bytes from 192.0.2.65: icmp_seq=2 ttl=64 time=1.3832 ms
68 bytes from 192.0.2.65: icmp_seq=3 ttl=64 time=4.2682 ms
68 bytes from 192.0.2.65: icmp_seq=4 ttl=64 time=1.1797 ms
68 bytes from 192.0.2.65: icmp_seq=5 ttl=64 time=1.2778 ms
Statistics: 5 sent, 5 received, 0% packet loss
```

Validating LDP Adjacency, Routes and Reachability

Run the following commands to show LDP neighbor and LDP session.

```
supervisor@rtbrick.net: op> show ldp neighbor
Instance: default
  Interface/Peer      LDP ID              Transport IP             Up Since                   Expires
  ifl-0/0/1/10     192.0.2.65:0        192.0.2.65               Thu Mar 21 06:27:38       in 14s
```

```
supervisor@rtbrick.net: op> show ldp session
Instance: default
  LDP ID            Peer IP                  State             Up/Down           FECRcvd   FECSent
  192.0.2.65:0      192.0.2.65               Operational       0d:00h:02m:56s          5         5
```

After configuring the LDP protocol, check the IPv4 labeled unicast routes, populated by LDP using the following command:

```
supervisor@rtbrick.net: op> show  route ipv4 labeled-unicast source ldp
Instance: default, AFI: ipv4, SAFI: labeled-unicast
Prefix/Label                             Source            Pref    Next Hop
Interface                    Label
192.0.2.65/32                            ldp               9       192.0.2.2                               ifl-
0/0/1/10               -
198.51.100.1/32                          ldp               9       192.0.2.2                               ifl-
0/0/1/10               10000
198.51.100.2/32                          ldp               9       192.0.2.2                               ifl-
0/0/1/10               10001
198.51.100.3/32                          ldp               9       192.0.2.2                               ifl-
0/0/1/10               10002
```

Ping the labeled unicast address 198.51.100.1 as follows:

```
supervisor@rtbrick.net: op> ping 198.51.100.1 instance default afi ipv4 safi labeled-unicas
t
68 bytes from 198.51.100.1: icmp_seq=1 ttl=64 time=.2344 ms
68 bytes from 198.51.100.1: icmp_seq=2 ttl=64 time=4.1493 ms
68 bytes from 198.51.100.1: icmp_seq=3 ttl=64 time=6.9031 ms
68 bytes from 198.51.100.1: icmp_seq=4 ttl=64 time=.2507 ms
68 bytes from 198.51.100.1: icmp_seq=5 ttl=64 time=3.2822 ms
Statistics: 5 sent, 5 received, 0% packet loss
```



The command argument `labeled-unicast` takes the ICMP requests through a labeled path while validating IP connectivity and hence, it prepends an MPLS label.

Validating BGP Adjacency, Routes and Reachability

Run the following commands to show BGP session and state.

```
supervisor@rtbrick.net: op> show bgp peer
Instance: default
  Peer                                Remote AS    State         Up/Down Time              PfxRcvd
PfxSent
  198.51.100.3                        4200000001   Established   0d:00h:04m:14s            1150000             5
```

After configuring BGP, check the IPv4 unicast routes, populated by BGP using the following command:

```
supervisor@rtbrick.net: op> show route ipv4 unicast source bgp instance default show route ipv4 unicast source bgp
instance default
Instance: default, AFI: ipv4, SAFI: unicast
Prefix/Label                            Source            Pref    Next Hop                                  ifl-
Interface
172.16.0.0/24                           bgp               200     198.51.100.2                              ifl-
0/0/1/10
172.16.1.0/24                           bgp               200     198.51.100.2                              ifl-
0/0/1/10
172.16.2.0/24                           bgp               200     198.51.100.2                              ifl-
0/0/1/10
172.16.3.0/24                           bgp               200     198.51.100.2                              ifl-
0/0/1/10
172.16.4.0/24                           bgp               200     198.51.100.2                              ifl-
0/0/1/10
172.16.5.0/24                           bgp               200     198.51.100.2                              ifl-
0/0/1/10
172.16.6.0/24                           bgp               200     198.51.100.2                              ifl-
0/0/1/10
172.16.7.0/24                           bgp               200     198.51.100.2                              ifl-
0/0/1/10
172.16.8.0/24                           bgp               200     198.51.100.2                              ifl-
0/0/1/10
172.16.9.0/24                           bgp               200     198.51.100.2                              ifl-
0/0/1/10
<...>
```

This command will list all the 1 million IPv4 BGP internet prefixes.

Pinging an IPv4 route (source: bgp) from the Multiservice Edge.

```
supervisor@rtbrick.net: op> ping 172.16.1.0
68 bytes from 172.16.1.0: icmp_seq=1 ttl=64 time=4.3976 ms
68 bytes from 172.16.1.0: icmp_seq=2 ttl=64 time=1.5301 ms
68 bytes from 172.16.1.0: icmp_seq=3 ttl=64 time=3.0536 ms
68 bytes from 172.16.1.0: icmp_seq=4 ttl=64 time=5.3049 ms
68 bytes from 172.16.1.0: icmp_seq=5 ttl=64 time=4.4273 ms
Statistics: 5 sent, 5 received, 0% packet loss
```

Check the IPv6 unicast routes, populated by BGP using the following command:

```
supervisor@rtbrick.net: op> show route ipv6 unicast source bgp instance defaultshow route ipv6 unicast source bgp
instance default
Instance: default, AFI: ipv6, SAFI: unicast
Prefix/Label                              Source           Pref    Next Hop
Interface
2004::/48                                 bgp              200     198.51.100.2                              ifl-
0/0/1/10
2004:0:1::/48                             bgp              200     198.51.100.2                              ifl-
0/0/1/10
2004:0:2::/48                             bgp              200     198.51.100.2                              ifl-
0/0/1/10
2004:0:3::/48                             bgp              200     198.51.100.2                              ifl-
0/0/1/10
2004:0:4::/48                             bgp              200     198.51.100.2                              ifl-
0/0/1/10
2004:0:5::/48                             bgp              200     198.51.100.2                              ifl-
0/0/1/10
2004:0:6::/48                             bgp              200     198.51.100.2                              ifl-
0/0/1/10
2004:0:7::/48                             bgp              200     198.51.100.2                              ifl-
0/0/1/10
2004:0:8::/48                             bgp              200     198.51.100.2                              ifl-
0/0/1/10
2004:0:9::/48                             bgp              200     198.51.100.2                              ifl-
0/0/1/10
2004:0:a::/48                             bgp              200     198.51.100.2                              ifl-
0/0/1/10

<...>
```

Pinging an IPv6 route (source: bgp) from the Multiservice Edge.

```
supervisor@rtbrick.net: op> ping 2004:0:1::
68 bytes from 2004:0:1::: icmp_seq=1 ttl=255 time=2.2185 ms
68 bytes from 2004:0:1::: icmp_seq=2 ttl=255 time=1.1854 ms
68 bytes from 2004:0:1::: icmp_seq=3 ttl=255 time=5.3917 ms
68 bytes from 2004:0:1::: icmp_seq=4 ttl=255 time=2.3068 ms
68 bytes from 2004:0:1::: icmp_seq=5 ttl=255 time=5.4419 ms
Statistics: 5 sent, 5 received, 0% packet loss
```

2.4. IPoE Subscriber Management Configuration

IP-over-Ethernet (IPoE) is an access technology that uses DHCP for IPv4 and DHCPv6 for IPv6 where both protocols are handled in the IPoE daemon (`ipoed`). IPoE subscribers are identified by IFP, VLANs and client MAC addresses.

The dynamic creation of IPoE subscribers is triggered by DHCPv4 discover or DHCPv6 solicit request from the subscriber. Response is postponed until the subscriber is successfully authenticated using the known authentication methods such as local or RADIUS, however authentication is not mandatory. After the authentication phase, IPv4/IPv6/IPv6-PD address is allocated to the subscriber either from the local pool or from RADIUS.

For IPoE Subscriber Management, the following configurations are mandatory:

1. Access Interface Configuration
2. Access Profile Configuration
3. AAA (Authentication, Authorization and Accounting) Profile Configuration. Based on the authentication requirement, configure any one of the following:
    a. Local Authentication
        i. Pool Configuration
        ii. User Profile Configuration
    b. RADIUS Authentication
        i. RADIUS Profile Configuration
        ii. RADIUS Server Configuration

This solution section discusses RADIUS authentication.

NOTES:

- Access interfaces can be configured without VLAN tags (untagged) and with one (single tagged) or two (double tagged) VLAN tags.
- There can be more than one interface configured for subscriber management and each interface can reference the same profile.

2.4.1. Configuring IPoE Subscriber Management

For detailed information about the subscriber configuration options, see the Subscriber Management Configuration Guide.

1. Configure the access profile `ipoe`.

```
set access access-profile ipoe
set access access-profile ipoe protocol dhcp enable true
set access access-profile ipoe protocol dhcp lease-time 60
set access access-profile ipoe protocol dhcpv6 enable true
set access access-profile ipoe protocol dhcpv6 lifetime 60
```

```
set access access-profile ipoe address-family ipv4 enable true
set access access-profile ipoe address-family ipv4 pool-name poolv4
set access access-profile ipoe address-family ipv4 instance default
set access access-profile ipoe address-family ipv4 dad-enable true
set access access-profile ipoe address-family ipv6 enable true
set access access-profile ipoe address-family ipv6 pool-name poolv6
set access access-profile ipoe address-family ipv6 prefix-delegation-pool-name poolv6pd
set access access-profile ipoe address-family ipv6 instance default
set access access-profile ipoe address-family ipv6 dad-enable true
commit
```

The access profile configuration is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show config access access-profile
{
  "rtbrick-config:access-profile": [
    {
      "profile-name": "ipoe",
      "protocol": {
        "dhcp": {
          "enable": "true",
          "lease-time": 60
        },
        "dhcpv6": {
          "enable": "true",
          "lifetime": 60
        }
      },
      "address-family": {
        "ipv4": {
          "enable": "true",
          "pool-name": "poolv4",
          "instance": "default",
          "dad-enable": "true"
        },
        "ipv6": {
          "enable": "true",
          "pool-name": "poolv6",
          "prefix-delegation-pool-name": "poolv6pd",
          "instance": "default",
          "dad-enable": "true"
        }
      }
    }
  ]
}
supervisor@rtbrick>multiservice-edge.rtbrick.net: op>
```

2. Configure the Authentication and Accounting (AAA) profile for `aaa-profile`.

```
set access aaa-profile aaa-profile
set access aaa-profile aaa-profile session-timeout 0
set access aaa-profile aaa-profile idle-timeout 0
set access aaa-profile aaa-profile aaa-radius-profile radius-profile
set access aaa-profile aaa-profile authentication order RADIUS
set access aaa-profile aaa-profile accounting order RADIUS
set access aaa-profile aaa-profile accounting interim-interval 86400
set access aaa-profile aaa-profile accounting ingress accounting-source POLICER
set access aaa-profile aaa-profile accounting egress accounting-source CLASS
commit
```

The access AAA configuration is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show config access aaa-profile
{
  "rtbrick-config:aaa-profile": [
    {
      "profile-name": "aaa-profile",
      "session-timeout": 0,
      "idle-timeout": 0,
      "aaa-radius-profile": "radius-profile",
      "authentication": {
        "order": "RADIUS"
      },
      "accounting": {
        "order": "RADIUS",
        "interim-interval": 86400,
        "ingress": {
          "accounting-source": "POLICER"        },
        "egress": {
          "accounting-source": "CLASS"
        }
      }
    }
  ]
}
supervisor@rtbrick>multiservice-edge.rtbrick.net: op>
```

3. Configure the access interface. Double-tagged interface is configured in this case as the access interface (`ifp-0/1/30`). The interface configuration assigns the access type, access profile, AAA profile, and further optional attributes like service-profile to the specified access interface.

```
set access interface double-tagged ifp-0/1/30 1001 1100 1001 1100
set access interface double-tagged ifp-0/1/30 1001 1100 1001 1100 access-type IPoE
set access interface double-tagged ifp-0/1/30 1001 1100 1001 1100 access-profile-name ipoe
set access interface double-tagged ifp-0/1/30 1001 1100 1001 1100 service-profile-name qos_service
set access interface double-tagged ifp-0/1/30 1001 1100 1001 1100 aaa-profile-name aaa-profile
set access interface double-tagged ifp-0/1/30 1001 1100 1001 1100 gateway-ifl lo-0/0/1/0
```

```
commit
```

The double-tagged access interface configuration is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show config access interface
{
  "rtbrick-config:interface": {
    "double-tagged": [
      {
        "interface-name": "ifp-0/1/30",
        "outer-vlan-min": 1001,
        "outer-vlan-max": 1100,
        "inner-vlan-min": 1001,
        "inner-vlan-max": 1100,
        "access-type": "IPoE",
        "access-profile-name": "ipoe",
        "service-profile-name": "qos_service",
        "aaa-profile-name": "aaa-profile",
        "gateway-ifl": "lo-0/0/1/0"
      }
    ]
  }
}
```

4. In this solution, we configure AAA authentication and accounting with RADIUS. To use RADIUS authentication and accounting both the RADIUS profile and RADIUS server configurations (see below) must be configured.

a. Configure RADIUS profile `radius-profile`.

```
set access radius-profile radius-profile
set access radius-profile radius-profile nas-ip-address 192.0.2.74
set access radius-profile radius-profile nas-port-format DEFAULT
set access radius-profile radius-profile nas-port-type Ethernet
set access radius-profile radius-profile authentication radius-server-profile-name radius
set access radius-profile radius-profile accounting radius-server-profile-name radius
commit
```

The RADIUS profile configuration is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show config access radius-profile radius-profile
{
  "rtbrick-config:radius-profile": [
    {
      "profile-name": "radius-profile",
      "nas-ip-address": "192.0.2.74",
      "nas-port-format": "DEFAULT",
      "nas-port-type": "Ethernet",
      "authentication": {
```

```
        "radius-server-profile-name": [
          "radius"
          ]
      },
      "accounting": {
        "radius-server-profile-name": [
          "radius"
          ]
      }
    }
  ]
}
supervisor@rtbrick>multiservice-edge.rtbrick.net: op>
```

b. Configure the RADIUS server `radius`.

```
set access radius-server radius
set access radius-server radius address 192.0.2.34
set access radius-server radius source-address 192.0.2.74
set access radius-server radius secret-plain-text testing123
set access radius-server radius routing-instance default
set access radius-server radius rate 300
set access radius-server radius authentication enable true
set access radius-server radius authentication retry 3
set access radius-server radius authentication timeout 5
set access radius-server radius accounting enable true
set access radius-server radius accounting timeout 30
set access radius-server radius coa enable true
commit
```



The attribute` secret-plain-text` will be converted to `secret-encrypted-text` in the show command output and value will be hashed.

The RADIUS server configuration is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config access radius-server radius
{
  "rtbrick-config:radius-server": [
    {
      "server-name": "radius",
      "address": "192.0.2.34",
      "source-address": "192.0.2.74",
      "secret-encrypted-text": "$2b2feb12f730107454b1be6a0f8242b0f",
      "routing-instance": "default",
      "rate": 300,
      "authentication": {
        "enable": "true",
        "retry": 3,
        "timeout": 5
      },
      "accounting": {
```

```
        "enable": "true",
        "timeout": 30
      },
      "coa": {
        "enable": "true"
      }
    }
  ]
}
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg>
```

5. Configure the IPv4 and IPv6 access pools.

```
set access pool poolv4
set access pool poolv4 ipv4-address low 203.0.113.1
set access pool poolv4 ipv4-address high 203.0.113.64
set access pool poolv6
set access pool poolv6 ipv6-prefix low 2001:db8:0:1::/64
set access pool poolv6 ipv6-prefix high 2001:db8:0:40::/64
set access pool poolv6pd
set access pool poolv6pd ipv6-prefix low 2001:db8:0:100::/56
set access pool poolv6pd ipv6-prefix high 2001:db8:0:4000::/56
commit
```

The access pool configuration is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config access pool
{
  "rtbrick-config:pool": [
    {
      "pool-name": "poolv4",
      "ipv4-address": {
        "low": "203.0.113.1",
        "high": "203.0.113.64"
      }
    },
    {
      "pool-name": "poolv6",
      "ipv6-prefix": {
        "low": "2001:db8:0:1::/64",
        "high": "2001:db8:0:40::/64"
      }
    },
    {
      "pool-name": "poolv6pd",
      "ipv6-prefix": {
        "low": "2001:db8:0:100::/56",
        "high": "2001:db8:0:4000::/56"
      }
```

```
    }
  ]
}
```

2.4.2. IPoE Quality of Service (QoS) Configuration



The QoS model explained in this document uses a complex HQoS model with the intent to showcase the complete range of QoS features available in RBFS. However, it may not be needed or desirable for all deployments. In such a case it should be possible to conceive of a simple QoS model as required by simplifying the provided QoS model.

Following are the steps involved in configuring and verifying IPoE QoS:

1. Configuring service profile to enable QoS on IPoE subscriber
2. Configuring downstream QoS
3. Configuring upstream QoS
4. Configuring QoS remarking
5. Configuring IPoE subscriber accounting for upstream and downstream traffic
6. Configuring IPoE subscribers QoS on BNG Blaster
7. Validating IPoE QoS on BNG Blaster

The figure below shows how QoS is configured for ingress and egress traffic.

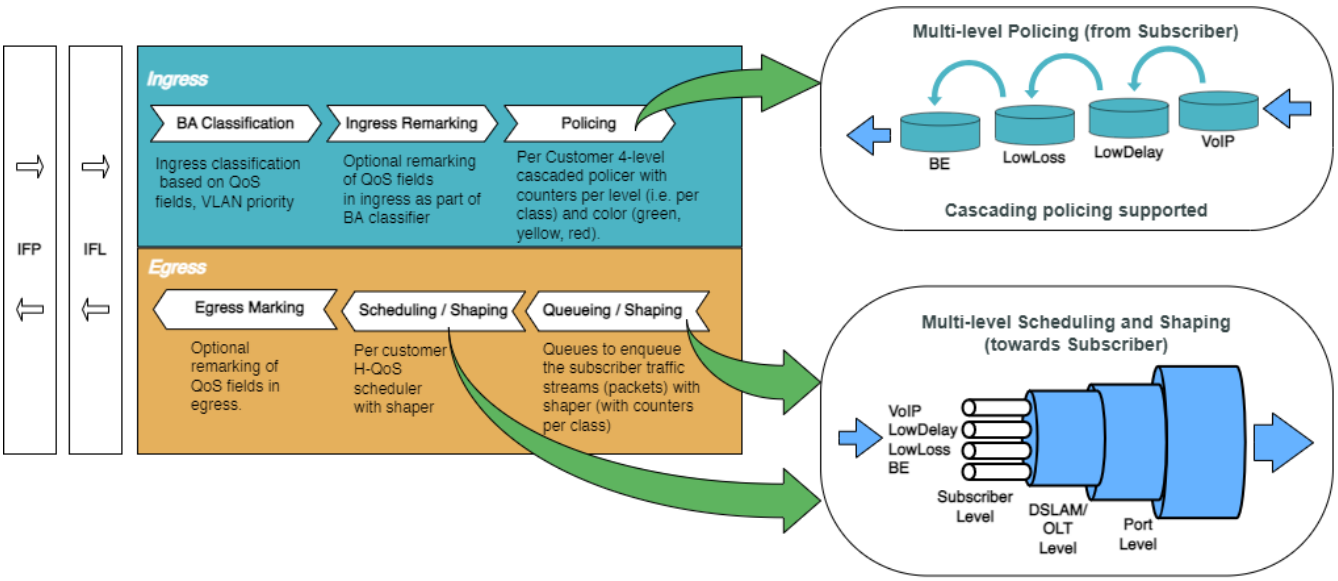


Figure 8. Hierarchical Quality of Service Primitives

For detailed information about the QoS configuration options, see the HQoS Configuration Guide.

Configure Service Profile

Service profile configuration in subscriber management allows to assign QoS configurations to a subscriber.

1. Configure the service profile to enable QoS. The service profile defined to enable Quality of Service with profile name is `residential`.

```
set access service-profile qos_service qos profile residential
commit
```

The configuration of the service profile named `residential` is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config access service-profile qos_service
{
  "rtbrick-config:service-profile": [
    {
      "profile-name": "qos_service",
      "qos": {
        "profile": "residential"
      }
    }
  ]
}
```

2. Enable QoS on IPoE subscriber access interface (`ifp-0/1/30`) to enable QoS for IPoE subscriber.

```
set access interface double-tagged ifp-0/1/30 1001 1100 1001 1100 service-profile-name qos_service
commit
```

Below is the double-tagged access interface on which the service profile `qos_service` is configured.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config access interface double-tagged ifp-0/1/30

{
  "rtbrick-config:double-tagged": [
    {
      "interface-name": "ifp-0/1/30",
      "outer-vlan-min": 1001,
      "outer-vlan-max": 1100,
      "inner-vlan-min": 1001,
      "inner-vlan-max": 1100,
      "access-type": "IPoE",
      "access-profile-name": "ipoe",
      "service-profile-name": "qos_service",
      "aaa-profile-name": "aaa-profile",
      "gateway-ifl": "lo-0/0/1/0"
    }
  ]
}
```

3. Configure QoS profile to enable on IPoE subscriber.

```
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 QoS Profile with all the primitives needed to enable traffic profiles on IPoE Subscribers is as follows:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config forwarding-options class-of-service profile
residential
{
  "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"
    }
  ]
}
```

Configure Downstream QoS

Downstream Quality of Service (QoS) is used to prioritize network traffic from the Internet to subscribers.

1. Enable global classification for downstream traffic.

```
set forwarding-options class-of-service global multifield-classifier-name global_mfc
commit
```

Below is the multi-field-classifier (MFC) based classifier for global enabling of downstream traffic classification.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config forwarding-options class-of-service global multifield-
classifier-name
{
  "rtbrick-config:multifield-classifier-name": "global_mfc"
}
```

2. Configure the MFC-based classifier with qualifiers and actions.

```
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
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc ordinal 1003
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc ordinal 1003 match ipv4-tos 192
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc ordinal 1003 match source-ipv4-
prefix 132.1.1.3/32
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc ordinal 1003 action forward-class
class-2
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc ordinal 1004
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc ordinal 1004 match ipv4-tos 224
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc ordinal 1004 match source-ipv4-
prefix 132.1.1.3/32
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc ordinal 1004 action forward-class
class-3
commit
```

The configuration of the QoS MFC-based Classifier for classification of downstream traffic from the core towards IPoE Subscriber is shown below.

```
supervisor@rtbrick>multiservice-edge.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"
            }
          },
          {
            "ordinal-value": 1003,
            "match": {
              "ipv4-tos": 192,
              "source-ipv4-prefix": "132.1.1.3/32"
            },
            "action": {
              "forward-class": "class-2"
            }
          },
          {
            "ordinal-value": 1004,
            "match": {
```

```
            "ipv4-tos": 224,
            "source-ipv4-prefix": "132.1.1.3/32"
          },
          "action": {
            "forward-class": "class-3"
          }
        }
      ]
    }
  ]
}
```

3. Enqueue classified traffic to different queues using class-to-queue mapping.

```
set forwarding-options class-of-service queue-group subs-4queues queue-numbers 4
set forwarding-options class-of-service class-queue-map subs-4queues
set forwarding-options class-of-service class-queue-map subs-4queues class class-0
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
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
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
set forwarding-options class-of-service class-queue-map subs-4queues class class-3 queue-name VO_SUBS
commit
```

Below is the QoS class-queue mapping configuration:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config forwarding-options class-of-service class-queue-map
subs-4queues class
{
    "rtbrick-config: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"
      }
    ]
  }
```

4. Configure the queues needed for enqueuing and dequeuing traffic streams.

```
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 LD_SUBS
```

```
set forwarding-options class-of-service queue LD_SUBS queue-size 625000
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 VO_SUBS
set forwarding-options class-of-service queue VO_SUBS queue-size 156250
set forwarding-options class-of-service queue VO_SUBS shaper-name shaper_VO
commit
```

The queue configuration is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config forwarding-options class-of-service queue
{
  "rtbrick-config: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",
    }
  ]
}
```

5. Configure the scheduler needed by Subscriber/Session scheduler-map and OLT scheduler-map.

```
set forwarding-options class-of-service scheduler pon0
set forwarding-options class-of-service scheduler pon0 type fair_queueing
set forwarding-options class-of-service scheduler pon0 shaper-name gpon-shaper
set forwarding-options class-of-service scheduler subs-4queues
set forwarding-options class-of-service scheduler subs-4queues shaper-name shaper_session
set forwarding-options class-of-service scheduler subs-4queues type strict_priority
set forwarding-options class-of-service scheduler subs-4queues composite false
commit
```

The configuration of the scheduler-map and OLT scheduler-map is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config  forwarding-options class-of-service scheduler
{
  "rtbrick-config:scheduler": [
    {
      "scheduler-name": "pon0",
      "shaper-name": "gpon-shaper",
```

```
      "type": "fair_queueing"
    },
    {
      "scheduler-name": "subs-4queues",
      "shaper-name": "shaper_session",
      "type": "strict_priority",
      "composite": "false"
    }
  ]
}
```

6. Configure the session/subscriber scheduler mapping for dequeuing traffic based on scheduler type for each queue:

```
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 QoS Subscriber/Session Scheduler-Map configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config forwarding-options class-of-service scheduler-map
subs-4queues-residential
{
```

```
  "rtbrick-config:scheduler-map": [
    {
      "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"
        }
      ]
    }
  ]
}
```

7. Configure the OLT scheduler-mapping for each PON to be scheduled according to the scheduler type.

```
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
commit
```

The OLT Scheduler-Map configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config forwarding-options class-of-service scheduler-map
schedmap-olt
{
  "rtbrick-config:scheduler-map": [
    {
      "scheduler-map-name": "schedmap-olt",
      "scheduler-name": [
```

```
        {
          "name": "pon0",
          "port-connection": "scheduler_to_port"
        }
      ]
    }
  ]
}
```

8. Configure downstream traffic shaping for both session schedulers and queues.



Queue Shaping is only on VO_SUBS Queue.

```
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
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 gpon-shaper
set forwarding-options class-of-service shaper gpon-shaper shaping-rate-high 2488000
set forwarding-options class-of-service shaper gpon-shaper shaping-rate-low 32000
commit
```

The shaping Configuration is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: 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
    },
    {
      "shaper-name": "gpon-shaper",
      "shaping-rate-high": 2488000,
      "shaping-rate-low": 32000
    }
  ]
}
```

Configure Upstream QoS

1. Configure the BA Classifier for the classification of multiple traffic streams targeted at IPoE subscribers:

```
set forwarding-options class-of-service classifier subs-pbit-class
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-802.1
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-802.1 codepoint 1
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-802.1 codepoint 1 class class-0
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-802.1 codepoint 1 remark-codepoint 7
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-802.1 codepoint 2
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-802.1 codepoint 2 class class-1
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-802.1 codepoint 2 remark-codepoint 7
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-802.1 codepoint 3
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-802.1 codepoint 3 class class-2
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-802.1 codepoint 3 remark-codepoint 7
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-802.1 codepoint 4
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-802.1 codepoint 4 class class-3
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-802.1 codepoint 4 remark-codepoint 7
commit
```

The configuration of the QoS BA-based Classifier for classification of upstream traffic towards IPoE Subscriber is shown below.

```
 supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config forwarding-options class-of-service classifier subs-
pbit-class
{
  "rtbrick-config: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
            }
          ]
        }
      ]
    }
  ]
}
```

2. Configure multi-level policer to police 4-Level traffic.

```
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 multi-level policer configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: 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"
    }
  ]
}
```

3. Map the classified traffic streams to different policer levels using class-to-policer mapping:

```
set forwarding-options class-of-service class-policer-map policer-map-residential
set forwarding-options class-of-service class-policer-map policer-map-residential class class-0
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
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
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
set forwarding-options class-of-service class-policer-map policer-map-residential class class-3 policer-level level-4
commit
```

The class-policer-map configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config forwarding-options class-of-service class-policer-map
policer-map-residential class
{
  "rtbrick-config: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"
    }
  ]
}
```

Configure QoS Remarking

1. Remark downstream traffic egressing from subscriber interface (egress remarking).

```
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 1
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-codepoint 1
color all
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-codepoint 1
color all remark-codepoint 6
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-codepoint 2
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-codepoint 2
color all
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-codepoint 2
color all remark-codepoint 6
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-codepoint 3
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-codepoint 3
color all
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-codepoint 3
color all remark-codepoint 6
```

```
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-codepoint 4
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-codepoint 4
color all
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-codepoint 4
color all remark-codepoint 6
commit
```

The remarking configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config forwarding-options class-of-service remark-map subs-
remarking-residential
{
  "rtbrick-config:remark-map": [
    {
      "remark-map-name": "subs-remarking-residential",
      "remark-type": [
        {
          "remark-type": "ieee-802.1",
          "match-codepoint": [
            {
              "match-codepoint": 1,
              "color": [
                {
                  "color": "all",
                  "remark-codepoint": 6
                }
              ]
            },
            {
              "match-codepoint": 2,
              "color": [
                {
                  "color": "all",
                  "remark-codepoint": 6
                }
              ]
            },
            {
              "match-codepoint": 3,
              "color": [
                {
                  "color": "all",
                  "remark-codepoint": 6
                }
              ]
            },
            {
              "match-codepoint": 4,
              "color": [
                {
                  "color": "all",
                  "remark-codepoint": 6
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

3. Remark upstream traffic ingressing to a subscriber's interface [ingress remarking]



In the upstream traffic classifier configuration shown below, remarking of all traffic streams with code point '7' is done.

```
set forwarding-options class-of-service classifier subs-pbit-class
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-802.1
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-802.1 codepoint 1
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-802.1 codepoint 1 class class-0
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-802.1 codepoint 1 remark-codepoint 7
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-802.1 codepoint 2
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-802.1 codepoint 2 class class-1
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-802.1 codepoint 2 remark-codepoint 7
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-802.1 codepoint 3
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-802.1 codepoint 3 class class-2
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-802.1 codepoint 3 remark-codepoint 7
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-802.1 codepoint 4
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-802.1 codepoint 4 class class-3
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-802.1 codepoint 4 remark-codepoint 7
commit
```

Below is the remarking configuration:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config forwarding-options class-of-service classifier subs-
pbit-class
{
  "rtbrick-config: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
            }
          ]
        }
      ]
    }
  ]
}
```

2.4.3. Configure Mirroring for Downstream Traffic Remark Validation

Configure Mirroring for Downstream Traffic

```
set forwarding-options mirror m1 source interface ifp-0/1/30
set forwarding-options mirror m1 source direction egress
set forwarding-options mirror m1 destination interface cpu-0/0/200
commit
```

Validate the downstream traffic remarking



This validation requires mirroring the subscriber access interface on the multiservice-edge device.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config forwarding-options mirror
{
  "rtbrick-config:mirror": [
    {
      "name": "m1",
      "destination": {
        "interface": "cpu-0/0/200"
      },
      "source": {
        "direction": "egress",
        "interface": "ifp-0/1/30"
      }
    }
  ]
}
```

The capture mirroring can be performed as shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> capture mirror

2023-04-03T13:58:59.651227+0000 e8:c5:7a:8f:76:f1 > 02:00:00:00:00:01, ethertype 802.1Q (0x8100), length 1022: vlan 1001,
p 6, ethertype 802.1Q, vlan 1001, p 6, ethertype IPv4, (tos 0xe0, ttl 63, id 0, offset 0, flags [DF], proto UDP (17),
length 1000)
    192.0.2.2.65056 > 203.0.113.1.65056: UDP, length 972

2023-04-03T13:58:59.651306+0000 e8:c5:7a:8f:76:f1 > 02:00:00:00:00:01, ethertype 802.1Q (0x8100), length 1022: vlan 1001,
p 6, ethertype 802.1Q, vlan 1001, p 6, ethertype IPv4, (tos 0xe0, ttl 63, id 0, offset 0, flags [DF], proto UDP (17),
length 1000)
    192.0.2.2.65056 > 203.0.113.1.65056: UDP, length 972
```

Configure Mirroring for Upstream Traffic Remark Validation

```
set forwarding-options mirror m1 source interface ifp-0/1/30
set forwarding-options mirror m1 source direction ingress
set forwarding-options mirror m1 destination interface cpu-0/0/200
```

```
commit
```

Validating the upstream traffic remarking



Mirror the core facing port on the Multiservice Edge device as shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config forwarding-options mirror m1
{
  "rtbrick-config:mirror": [
    {
      "name": "m1",
      "destination": {
        "interface": "cpu-0/0/200"
      },
      "source": {
        "direction": "ingress",
        "interface": "ifp-0/1/30"
      }
    }
  ]
}
```

The capture mirroring can be performed as shown below. It confirms all four traffic streams noted with codepoint=7.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> capture mirror

2023-04-03T13:53:31.400011+0000 02:00:00:00:00:01 > e8:c5:7a:8f:76:f1, ethertype 802.1Q (0x8100), length 1022: vlan 1001,
p 2, ethertype 802.1Q, vlan 1001, p 2, ethertype IPv4, (tos 0xa0, ttl 64, id 0, offset 0, flags [DF], proto UDP (17),
length 1000)
    203.0.113.1.65056 > 192.0.2.2.65056: UDP, length 972

2023-04-03T13:53:31.400167+0000 02:00:00:00:00:01 > e8:c5:7a:8f:76:f1, ethertype 802.1Q (0x8100), length 1022: vlan 1001,
p 3, ethertype 802.1Q, vlan 1001, p 3, ethertype IPv4, (tos 0xc0, ttl 64, id 0, offset 0, flags [DF], proto UDP (17),
length 1000)
    203.0.113.1.65056 > 192.0.2.2.65056: UDP, length 972
```

2.4.4. Configure FreeRADIUS Server

Installation of FreeRADIUS

FreeRADIUS server can be installed on any Linux OS distribution. For information about installing FreeRADIUS, see Installing the FreeRADIUS Server.

Remove the Unsupported Files

It is necessary to remove `echo`, `ntlm_auth`, `eap`, `echo`, and `mschap` files once FreeRadius has been installed, since they are not required by this reference design. To remove these files, enter the following commands:

```
rm -rf /etc/freeradius/3.0/mods-enabled/echo
rm -rf /etc/freeradius/3.0/mods-enabled/ntlm_auth
rm -rf /etc/freeradius/3.0/mods-enabled/eap
rm -rf /etc/freeradius/3.0/mods-enabled/echo
rm -rf /etc/freeradius/3.0/mods-enabled/mschap
```

FreeRADIUS User and Group Settings

Using the following commands, one can set the user or group as root.

```
sed -i 's/User=freerad/User=root/g' /lib/systemd/system/freeradius.service
sed -i 's/Group=freerad/Group=root/g' /lib/systemd/system/freeradius.service
```

Run the command for reloading 'systemd' after user or group settings:

```
systemctl daemon-reload
```

Configure the FreeRADIUS Files

It is necessary to configure the FreeRADIUS files once FreeRadius has been installed. The configuration files can be found under `/etc/freeradius/3.0/`.

authorize

Using the following command, one can view the `authorize` file in its default location.

```
~:/etc/freeradius/3.0 # cat mods-config/files/authorize
```

Replace the content of the `authorize` file with the following:

```
$INCLUDE /etc/freeradius/3.0/ipoe_users_file
```



This file can also be downloaded from the appendix (Appendix C: RADIUS Server Configuration).

ipoe_users_file

The IPoE Users file (`ipoe_users_file`) mentioned in the above section includes subscriber profile parameters as shown below.

```
"02:00:00:00:00:01@ipoe" Cleartext-Password := "ipoe"
    Service-Type = Framed-User,
```

```
    Class = IPOE,
    Framed-IP-Address = 203.0.113.1,
    Framed-IP-Netmask = 255.255.255.255,
    RtBrick-DNS-Primary = 203.0.113.200,
    RtBrick-DNS-Secondary  = 203.0.113.201,
    Framed-IPv6-Prefix = 2001:db8:0:1::1/128,
    Delegated-IPv6-Prefix = 2001:db8:0:3::/64,
    RtBrick-DNS-Primary-IPv6  = 2001:db8:0:20::1,
    RtBrick-DNS-Secondary-IPv6 = 2001:db8:0:20::2,
    Session-Timeout = 0,
    Idle-Timeout = 0,
    Reply-Message = "FOOBAR Internet",
    Acct-Interim-Interval = 120,
    RtBrick-QoS-Profile = "residential"

DEFAULT User-Name =~ '^[0-9a-f\:]+@ipoe$', Cleartext-Password := 'ipoe'
    Service-Type = Framed-User,
    Class = "IPOE",
    Acct-Interim-Interval = 900
```

The `ipoe_users_file` can be created with the above content in the `/etc/freeradius/3.0/` path. Alternatively, this file can be downloaded from the appendix section of this guide and placed at `/etc/freeradius/3.0/`.

clients.conf

Clients.conf file shall be configured with the expected RADIUS client IP address and secret.

```
~:/etc/freeradius/3.0 # cat clients.conf
client rtbrick {
        ipaddr          = 192.0.2.74
        secret          = testing123
        shortname       = rtbrick
        nas_type        = other
        require_message_authenticator = no
}
```

The `clients.conf` file (/etc/freeradius/3.0/clients.conf) used for this reference design can be downloaded from the appendix section of this guide.

radiusd.conf

The `radiusd.conf` file should be configured with the expected RADIUS authentication and accounting parameters.

```
prefix = /usr
exec_prefix = /usr
```

```
sysconfdir = /etc
localstatedir = /var
sbindir = ${exec_prefix}/sbin
logdir = /var/log/freeradius
raddbdir = /etc/freeradius
radacctdir = ${logdir}/radacct
name = freeradius
confdir = ${raddbdir}
modconfdir = ${confdir}/mods-config
run_dir = ${localstatedir}/run/${name}
db_dir = ${raddbdir}
libdir = /usr/lib/freeradius
pidfile = ${run_dir}/${name}.pid
correct_escapes = true
max_request_time = 5
cleanup_delay = 0
max_requests = 16384
hostname_lookups = no
log {
    destination = files
    file = ${logdir}/radius.log
    stripped_names = no
    auth = yes
}
checkrad = ${sbindir}/checkrad
security {
    # user = radius
    # group = radius
    allow_core_dumps = no
    max_attributes = 200
    reject_delay = 0
    status_server = no
@openssl_version_check_config@
}
proxy_requests  = no
$INCLUDE clients.conf
thread pool {
    start_servers = 32
    max_servers = 128
    min_spare_servers = 8
    max_spare_servers = 16
    max_queue_size = 16384
    max_requests_per_server = 0
    auto_limit_acct = no
}
modules {
    $INCLUDE mods-enabled/
}
instantiate {
    files
    linelog
}
```

```
server default {
    listen {
        type = auth
        ipaddr = *
        port = 1812
    }
    listen {
        type = acct
        ipaddr = *
        port = 1813
    }
    authorize {
        update request {
            FreeRADIUS-Client-Shortname = "%{&request:Client-Shortname}"
        }
        if (&request:Client-Shortname == "rtbrick-server") {
            rtbrick-server-log
        }
        files
        pap
    }
    authenticate {
        pap
    }
    post-auth {
        if (&request:Client-Shortname == "rtbrick-server") {
            rtbrick-server-log
        }
        Post-Auth-Type REJECT {
            update reply {
                Reply-Message := "Login Failed. Please check your username and password."
            }
            attr_filter.access_reject
        }
    }
    preacct {
        ok
    }
    accounting {
        if (&request:Client-Shortname == "rtbrick-server") {
            rtbrick-server-log
        }
    ok
    }
    session {
    }
}
```

Radiusd.conf should be configured to use UDP ports 1812 and 1813 for authentication and accounting, respectively. Additionally, `rtbrick-server-log` should be added to the parameters for authorise, authenticate, and accounting.

The `radiusd.conf` file (/etc/freeradius/3.0/radiusd.conf) used for this reference design can be downloaded from the appendix section (Appendix C) of this guide.

detail

The `detail` file shall be configured for the RADIUS accounting logs.

```
/etc/freeradius/3.0 # cat mods-enabled/detail
    permissions = 0666detail rtbrick-server-log {
    filename = ${radacctdir}/rtbrick-server-detail.log

    header = "%t;%{NAS-IP-Address};%I;%{Packet-Src-Port}"
    log_packet_header = no
}
```

Ensure that `rtbrick-server-log` is specified in the `detail` file.

The `detail` file (/etc/freeradius/3.0/mods-enabled/detail) used for this reference design can be downloaded from the appendix section (Appendix C) of this guide.

dictionary.rtbrick

Add the RtBrick RADIUS dictionary (dictionary.rtbrick) to `/usr/share/freeradius/dictionary.rtbrick` and include it in` /usr/share/freeradius/dictionary`.

The `dictionary.rtbrick` contains the RBFS attributes in FreeRADIUS format.

- Click here to download the `radius_config.zip`, which contains the `dictionary.rtbrick` file.

Stopping and Starting the FreeRADIUS Server for any Changes

For any changes, stop and restart the FreeRADIUS server.

To stop the server, enter the following command:

```
sudo service freeradius stop
```

To start the server, enter the following command:

```
sudo service freeradius start
```

The FreeRadius server is now ready to provide AAA (Authentication, Accounting & Authorization) services to logging in subscribers.

Displaying the RADIUS Service Status

Run the following command to determine whether RADIUS service is active:

```
sudo service freeradius status
```

2.4.5. Validating IPoE Subscriber Bring-Up

Using traffic streams on both upstream and downstream directions with traffic packets and bytes statistics, IPoE Subscriber sessions can be "ESTABLISHED".

The validation can be performed in two steps:

1. Establishing the IPoE subscriber
2. Pinging the subscriber IPv4/IPv6 address

BNG Blaster - IPoE Subscribers with Traffic Streams

Using BNG Blaster, which emulates IPoE clients, session traffic("session-traffic") and streams traffic with different code-points("streams"), one can test IPoE subscriber management feature.

The data traffic can be defined in Blaster by configuring session/streams traffic.

- Session Traffic in Blaster can be enabled by specifying `"autostart": true`. Once the session is established, traffic starts automatically. Eventually if you want to stop or start the traffic (session or streams), press F7/F8.
- Streams traffic in Blaster can be enabled by specifying `stream-group-id` at both streams (`"streams":`) and access interface levels (`interfaces:access`).

For information about using BNG Blaster, see Starting BNG Blaster.

Validating the IPoE Session on BNG Blaster

To validate the IPoE Session on BNG Blaster, switch to the Service Node. On the BNG Blaster terminal, the count of the "Established" session will be 1. Also, the logging of the same terminal will display "ALL SESSIONS ESTABLISHED" and "ALL SESSION TRAFFIC FLOWS VERIFIED" as highlighted in the below image.

In the image below, one can see the details of the established Single Subscriber session.

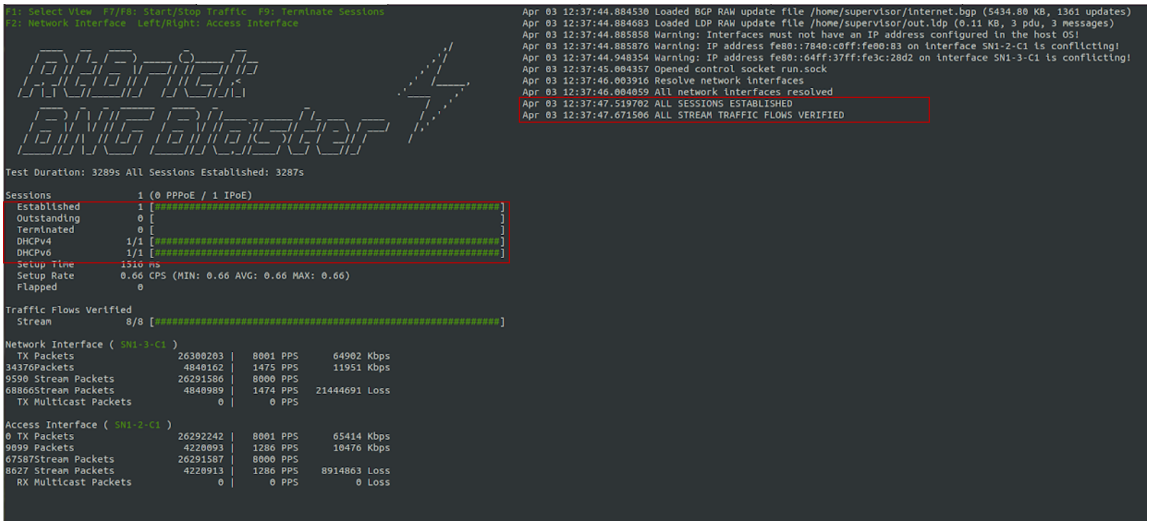


Figure 9. Validating the IPoE Session on BNG Blaster

Fig 3: BNG Blaster terminal view

Visit the following URL for more information on BNG Blaster:

https://rtbrick.github.io/bngblaster/

Viewing the Subscribers and the Subscriber Details on RBFS

Enter the following command to view the list of subscribers.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show subscriber
Subscriber-Id           Interface         VLAN      Type   State
216454257090494470      ifp-0/1/30        1001:1001 IPoE   ESTABLISHED
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg>
```

Enter the following command to view the details of the subscriber with ID 216454257090494470.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show subscriber 216454257090494470 detail
Subscriber-Id: 216454257090494470
    Type: IPoE
    State: ESTABLISHED
    Created: Mon Apr 03 13:51:17 GMT +0000 2023
    Interface: ifp-0/1/30
    Outer VLAN: 1001
    Inner VLAN: 1001
    Client MAC: 02:00:00:00:00:01
    Server MAC: e8:c5:7a:8f:76:f1
    IFL: ipoe-0/1/30/216454257090494470
    Username: 02:00:00:00:00:01@ipoe
    Access-Profile: ipoe
    AAA-Profile: aaa-profile
    Service-Profile: qos_service
    Reply-Message: FOOBAR Internet
    Session-Timeout: 0 (disabled)
    Idle-Timeout: 0 (disabled)
    MTU: 1500 Profile: N/A
```

```
    IPv4:
        Instance: default
        Address: 203.0.113.1/255.255.255.255
        Address Active: True
        Primary DNS: 203.0.113.200
        Secondary DNS: 203.0.113.201
    IPv6:
        Instance: default
        RA Prefix: 2001:db8:0:1::1/128
        RA Prefix Active: True
        Delegated Prefix (DHCPv6): 2001:db8:0:3::/64
        Delegated Prefix Active: True
        Primary DNS: 2001:db8:0:20::1
        Secondary DNS: 2001:db8:0:20::2
    Accounting:
        Session-Id: 216454257090494470:1680529877
        Start-Time: 2023-04-03T13:51:18.543868+0000
        Interims Interval: 120 seconds
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg>
```

Pinging the Subscriber (source: IPOE) from Multiservice Edge

Before pinging a subscriber, use the `show route <···>` command to display the subscriber IPs at the Multiservice Edge.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show route ipv4 unicast source ipoe
Instance: default, AFI: ipv4, SAFI: unicast
Prefix/Label                            Source           Pref    Next Hop                                Interface
203.0.113.1/32                          ipoe             7       -                                       ipoe-
0/1/30/216454257090494470
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg>
```

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show route ipv6 unicast source ipoe
Instance: default, AFI: ipv6, SAFI: unicast
Prefix/Label                            Source           Pref    Next Hop                                Interface
2001:db8:0:1::1/128                     ipoe             7       -                                       ipoe-
0/1/30/216454257090494470
2001:db8:0:3::/64                       ipoe             7       fe80::ffff:ffff:ff00:1                  ipoe-
0/1/30/216454257090494470
```

From the above list, ping `203.0.113.1` as shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> ping 203.0.113.1
68 bytes from 203.0.113.1: icmp_seq=1 ttl=64 time=1.3463 ms
68 bytes from 203.0.113.1: icmp_seq=2 ttl=64 time=10.5234 ms
68 bytes from 203.0.113.1: icmp_seq=3 ttl=64 time=9.8053 ms
68 bytes from 203.0.113.1: icmp_seq=4 ttl=64 time=11.0041 ms
68 bytes from 203.0.113.1: icmp_seq=5 ttl=64 time=10.9300 ms
Statistics: 5 sent, 5 received, 0% packet loss
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg>
```

Validating Traffic Streams

Traffic streams can be used to perform various forwarding verifications.

For upstream traffic capture, enter the following command:

```
capture interface ifp-0/1/30 direction in
```

For downstream traffic capture, enter the following command:

```
capture interface ifp-0/1/30 direction out
```

Here, `ifp-0/1/30` refers to the access interface.

Validating the IPoE QoS on BNG Blaster

To validate the IPoE QoS on BNG Blaster, switch to the Service Node. Navigate to the Streams and Session Traffic terminal by pressing F1 function key in your keyboard.

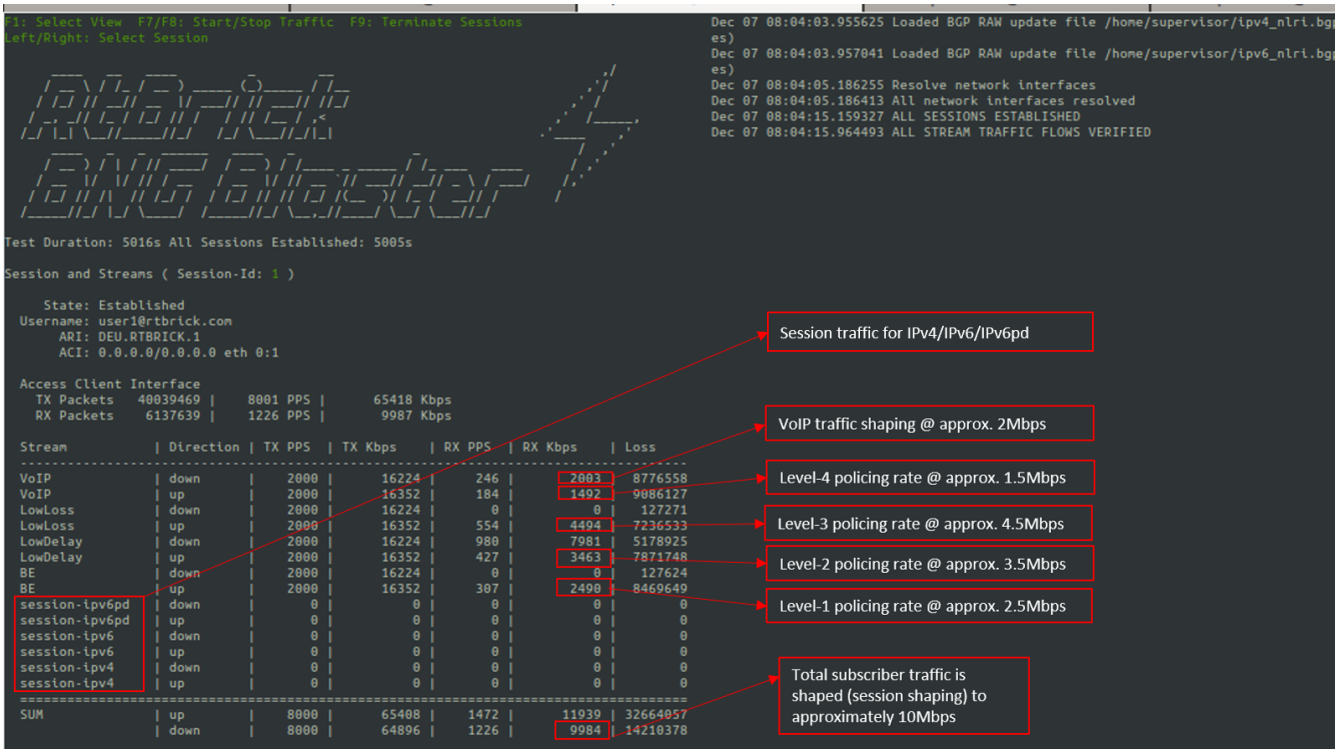


Figure 10. Reading Output from BNG Blaster

As shown in the above image, the VoIP downstream traffic has been shaped (session shaping) to 2Mbps. Similarly, the total subscriber traffic has been shaped approximately to 10Mbps.

Following are the upstream traffic rates of different policer levels:

- Level-1 Rates ~=2.5Mbps
- Level-2 Rates ~=3.5Mbps
- Level-3 Rates ~=4.5Mbps

- Level-4 Rates ~=1.5Mbps

2.4.6. IPoE Subscriber Accounting for Upstream and Downstream Traffic

Run the "show subscriber" command to view the list of subscribers.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show subscriber
Subscriber-Id          Interface        VLAN      Type   State
216454257090494470     ifp-0/1/30       1001:1001 IPoE   ESTABLISHED
ESTABLISHEDsupervisor@rtbrick>multiservice-edge.rtbrick.net: cfg>
```

Specify the Subscriber-ID to find the specific subscriber's accounting details:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show subscriber 216454257090494470 accounting
Subscriber-Id: 216454257090494470
    IFL: ipoe-0/1/30/216454257090494470
    Start Timestamp: Mon Apr 03 13:51:18 GMT +0000 2023
    Idle Timestamp: Mon Apr 03 13:51:38 GMT +0000 2023
    Session-Timeout: 0 seconds
    Idle-Timeout: 0 seconds
    Session Statistics:
        Ingress: 28185 packets 27339450 bytes
        Egress: 23990 packets 23990000 bytes
    LIF Statistics:
        Ingress: 0 packets 0 bytes
        Egress: 0 packets 0 bytes
    Egress Class (Queue) Statistics:
        class-0: 4 packets 4000 bytes dropped: 0 packets 0 bytes
        class-1: 19191 packets 19191000 bytes dropped: 0 packets 0 bytes
        class-2: 3 packets 3000 bytes dropped: 0 packets 0 bytes
        class-3: 4792 packets 4792000 bytes dropped: 0 packets 0 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: 5819 packets 5644430 bytes dropped: 31270 packets 30331900 bytes
        Level 2: 8170 packets 7924900 bytes dropped: 29029 packets 28158130 bytes
        Level 3: 10540 packets 10223800 bytes dropped: 26638 packets 25838860 bytes
        Level 4: 3656 packets 3546320 bytes dropped: 33614 packets 32605580 bytes
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg>
```

2.4.7. Configuring Lawful Intercept (LI)



This section is still a work in progress.

2.5. Appendixes

Appendix A: RBFS Multiservice Edge Configuration

The RBFS Multiservice Edge configuration file (`multiservice-edge.json`) can be downloaded from here.

- Click here to download the RBFS Multiservice Edge configuration file.

Appendix B: TACACS+ Server Configuration

The TACACS+ server configuration file (`tac_plus.conf`) can be downloaded from here.

- Click here to download the TACACS+ server configuration file.

Appendix C: RADIUS Server Configuration

The RADIUS server configuration files (`radius_config.zip`) can be downloaded from here. The zip archive contains the set of configuration files needed to configure the RADIUS server.

- Click here to download the .zip file that contains the RADIUS server configuration files.

Appendix D: BNG Blaster Configuration

The BNG Blaster configuration file (`blaster.json`) can be downloaded from here.

- Click here to download the BNG Blaster configuration files.

The JSON file (`ospf_3node.json`) which is used to simulate R-1, R-2, and RR on BNG Blaster, can be downloaded from here.

- Click here to download the `ospf_3node.json` file.

3. RBFS Multiservice Edge with CGNAT for IPoE and PPPoE Subscribers

3.1. RBFS Multiservice Edge with CGNAT for IPoE and PPPoE Subscribers

3.1.1. Overview

This document provides information for validating CGNAT (Carrier-Grade Network Address Translation) implementation using the CGNAT-enabled RBFS Multiservice Edge device. It depicts the implementation that uses both IPoE and PPPoE subscriber traffic and IPv4 network address translation in the RBFS Multiservice Edge deployed at the ISP network.

The document presents a use-case scenario that shows 10,000 single Q subscribers (8000 IPoE subscribers and 2000 PPPoE subscribers) on the same port. It implements a QoS profile that enables queueing, scheduling, and shaping for downstream traffic and single-level policing for upstream traffic per subscriber. The implementation uses a default port-block size 32 for address translation and OSPFv2/v3 protocols for route distribution. The default port block size cannot be changed.

The document presents a single use case scenario and configuration information required to validate the use case. The guide provides configuration information for general platform settings, Carrier-Grade Network Address Translation, various Access and Routing Protocols, Subscriber Management, and Quality of Service (QoS). The document provides information specifically on how to validate this particular implementation and for more information on any specific application, refer to https://documents.rtbrick.com/.

This guide is not intended to be an exhaustive guide of all RBFS features and does not provide information on features such as Multicast, Lawful Intercept, and so on.

Currently, the scope of the guide is limited to the basic features and configurations for validation purposes. This guide does not provide information about the advanced RBFS features such as multicast.

3.1.2. About RBFS Multiservice Edge

The RtBrick Multiservice Edge is delivered as a container, running on RtBrick Host provided by the hardware ODM manufacturers. Platforms that support Multiservice Edge include EdgeCore CSR440, Edgecore AGR420, UfiSpace S9510-28DC, UfiSpace S9600-72XC, and UfiSpace S9600-102XC. The RtBrick Multiservice Edge software runs on powerful bare-metal switches as an open BNG.

The BNG is designed to dynamically deliver the following services:

1. Discovering and managing subscriber sessions for IPoE/PPPoE subscribers
2. Providing authentication, authorization and accounting (AAA)

3.1.3. About BNG Blaster

BNG Blaster is an open-source network tester developed by RtBrick to validate the performance and scalability of BNG devices. The BNG blaster can emulate both subscribers at scale and network elements such as routers. Therefore, it is possible to create complex topologies with just the DUT and a server that hosts the BNG Blaster, thereby minimizing the equipment needed to test the DU. For more information, refer to the BNG Blaster documentation. To use BNG Blaster, download and install the BNG Blaster software onto the system that meets the tool's system requirements. The software can be downloaded from https://github.com/rtbrick/bngblaster/releases.

3.1.4. About RBFS Carrier-Grade NAT

Both BNG and CGNAT functionalities coexist in a CGNAT-enabled RBFS platform. RBFS CGNAT supports NAT44 functionality that maps IPv4 subscriber private addresses to IPv4 public addresses. RBFS CGNAT translates the source private IPv4 address and port number to a public source IPv4 address and port. Multiple private source IPv4 addresses can be translated into a single public IPv4 address, and the unique port numbers help to identify the private network devices from each other.

Address Translation for Upstream Traffic

When a device with a private IP address initiates communication with a destination device on the Internet, RBFS CGNAT chooses an available public IP address from the pool and creates an entry in its translation table. When packets from the private IP address traverse the network toward the internet, the CGNAT device translates the source IP address of these packets from the private address to the corresponding public address based on the entries in the translation table. It also performs port translation to maintain uniqueness, as multiple private addresses are mapped to the same public address.

It performs the address translation by replacing the source private IPv4 address with the public IPv4 address and the source port number (TCP/UDP) with the unique port number.

Reverse Address Translation for Downstream Traffic

When downstream traffic is received from servers on the Internet, the CGNAT device reverses the translation by replacing the destination public IP address with the corresponding private IP address from its translation table and the unique port number with the original port number.

3.1.5. RBFS Multiservice Edge with CGNAT Implementation Architecture

When your device with a private IP address sends a request to the destination device, CGNAT assigns to the packets a unique port number and translates the private IPv4 address of the device into the public IPv4 address.

The following diagram illustrates the architecture of Multiservice Edge with CGNAT that serves routing and address translation functions between a private network and a public network.

About this topology:

1. The Service Node is an Ubuntu server on which a new container is spawned with the associated interfaces for running BNG Blaster tests.

2. BNG Blaster emulates both the routing and access functions. It is an open-source network tester developed by RtBrick to validate the performance and scalability of BNG devices. The BNG blaster can emulate both subscribers at scale and network elements such as routers. Therefore, it is possible to create complex topologies with just the DUT and a server that hosts the BNG Blaster, thereby minimizing the equipment needed to test the DU. For more information, refer to the BNG Blaster documentation. To use BNG Blaster, download and install the BNG Blaster software onto the system that meets the tool's system requirements. The software can be downloaded from https://github.com/rtbrick/bngblaster/releases.
3. The topology emulates PPPoE and IPoE subscribers and traffic between RBFS switch and the core network.
4. In this topology, OSPFv2/v3 is used for internal routing within the AS, LDP is used to build the MPLS underlay for label switching, and iBGP is used to learn Internet prefixes and exchange routing information within the AS. These protocols establish route reachability.
5. The objective of this topology is to demonstrate complete RBFS Carrier-Grade Network Address Translation for IPoE and PPPoE subscribers.
6. R-1 and R-2 are simulated using BNG Blaster.

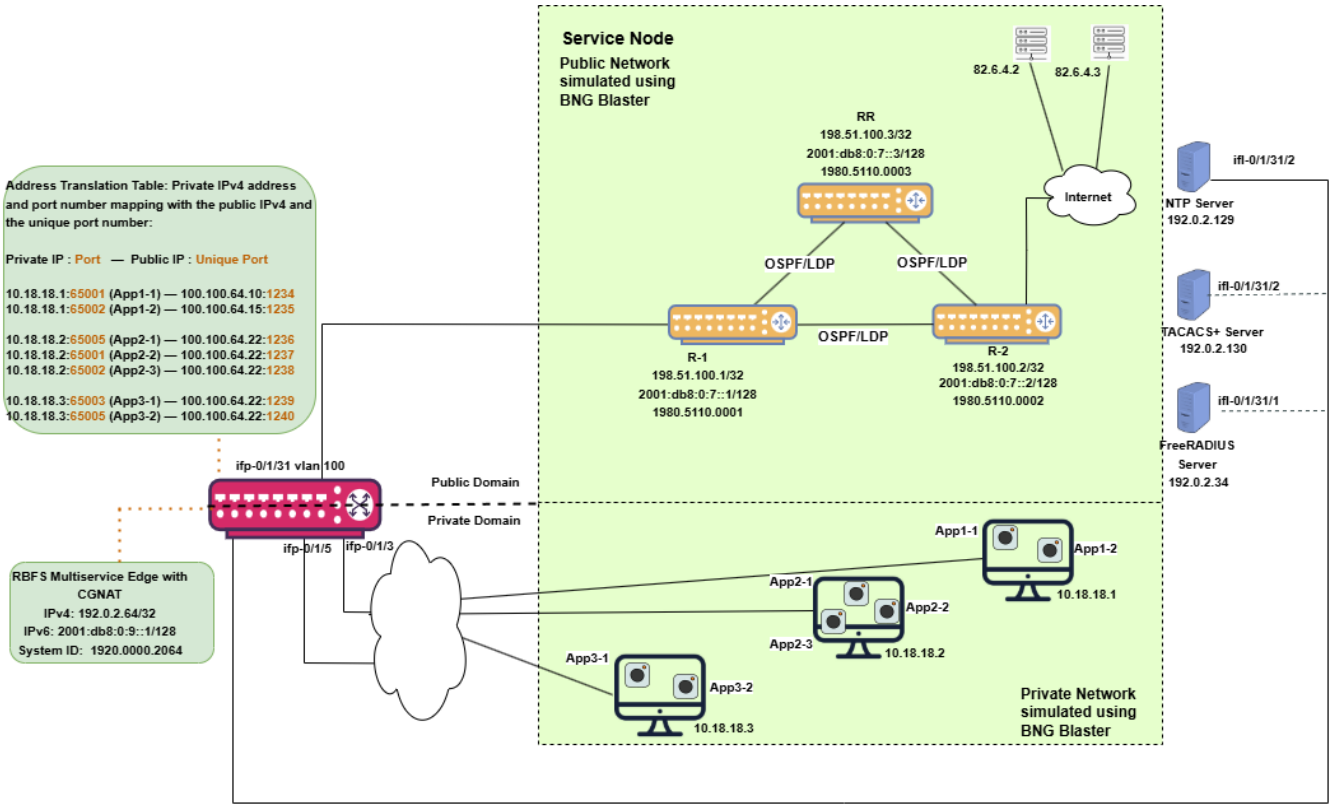


Figure 11. RBFS Multiservice Edge with CGNAT Implementation

At a high level, the diagram shows a private network and the public network which are simulated using BNG Blaster. Three host devices in a private network send traffic to servers on the public network. The host devices, that run two or three applications such as `App1-1`, `App2-2`, and so on, send traffic to servers on the Internet.

The private and public networks are simulated using BNG Blaster.

The PPPoE and IPoE subscriber traffic received by the CGNAT device interfaces `ifp-0/1/3` and `ifp-0/1/5`. The CGNAT-enabled RBFS performs address translation and sends the packets to the

destination.

The address translation table shows the mapping between the private IPv4 address (plus the port number) and the public IPv4 address (plus the unique port number).

For example, the private IPv4 address `10.18.18.1` and port number `65001` (App1-1) is mapped to the public IPv4 `100.100.64.10` and port number `1234`. CGNAT performs address translation from private IPv4 address to public IPv4 address upstream traffic and a reversal address translation from public IPv4 address to private IPv4 address for downstream traffic.

RBFS CGNAT maintains an Address Translation Table that records the mapping between the private IPv4 address and the public IPv4 address (with the port numbers).

It performs the address translation by replacing the source private IPv4 address with the public IPv4 address and the source port number (TCP/UDP) with the unique port number. The private IPv4 address and port number and the mapped public IPv4 address and the port number are recorded in the Address Translation Table. After the address translation, the traffic is forwarded to the R-1 router and finally to the destination servers in the public network.

A reversal address translation happens for the downstream traffic when the destination servers send packets to the RBFS device. After receiving the downstream packets, the CGNAT device examines the translation table mapping and performs a reversal address translation. It replaces the public IPv4 with the respective private IPv4 addresses and port numbers.

3.1.6. Deploying RBFS Multiservice Edge

RBFS Multiservice Edge runs on small form-factor hardware that allows deployments in street-site cabinets. The `rtbrick-toolkit` is a meta package that can be used to install all the tools needed to work with RBFS images (container or RtBrick Host installer) and the RBFS APIs. For more information, see RBFS Installation Guide.

3.1.7. Using the RBFS CLI

Connect to the `Multiservice Edge` node.

```
$ ssh <multiservice-edge-management-ip> -l supervisor
supervisor@<multiservice-edge-management-ip>'s password:
```

The password for `multiservice-edge-management-ip` should be entered here.

As a result, the CLI prompt will look like this:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net:~ $
```

Open the RBFS CLI.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net:~ $ cli
```

The CLI has three different modes:

- `operation` mode is a read-only mode to inspect and analyse the system state
- `config` mode allows modifying the RBFS configuration
- `debug` mode provides advanced tools for trouble-shooting

The `switch-mode` command allows switching between the different modes. The `show` commands allow inspecting the system state. The `set` and `delete` commands, which are only available in the configuration mode, allow modifying or deleting the current configuration. The `commit` command persists the changes. RBFS provides a commit history that allows reviewing changes (`show commit log`) and restoring a previous configuration (`rollback`). There are also commands to `ping` destinations, `capture` network traffic, `save` the configuration to or `load` the configuration from a file.

The CLI supports abbreviating commands, provides suggestions by hitting the `[tab]` key and displays a context help by entering `?`.

For more information on how to use the RBFS CLI, see the RBFS CLI User Guide.

3.2. Configuration and Settings

3.2.1. Platform Configuration and Settings

This section provides information about the platform and how to set various required configurations for the platform.

Know your Device

The configurations provided in this document are generated on the UfiSpace S9600-72XC platform. 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 [1GbE/10GbE/25GbE] and 8x100GE [40GbE/100GbE] high-speed ports with a switching capacity of up to 2.4Tbs.

The RBFS Multiservice Edge software is installed on the UfiSpace S9600-72XC.



Although the device used here is UfiSpace S9600-72XC, the configuration will stay exactly the same for any other device that supports the Multiservice Edge image.

For more information about the hardware specifications of UfiSpace S9600-72XC, see Hardware Specification.

Prerequisites

- Access to BNG Blaster, an open-source network testing platform for access and routing protocols. For information on obtaining and building BNG Blaster, see https://rtbrick.github.io/bngblaster/.
- Access to FreeRADIUS, a free RADIUS suite. For accessing FreeRADIUS, see https://freeradius.org/.

- Access to Syslog server.

Restore Configuration

Depending on the deployment scenario, a running configuration can be applied or restored as needed.

To enable configuration restore, enter the `set system load-last-config true` command as shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> set system load-last-config
true
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> commit
```

For more information, see the section "Running Configuration" of the RBFS NOC Troubleshooting Guide.

3.2.2. General Configuration

To enable testing, some basic primitives need to be configured. These general configurations include loopback interface for identifying and accessing the device on network, NTP for setting accurate time across a whole network of devices, TACACS+ for user authentication, user management for user configuration, license for accessing RBFS, Resmon for resource monitoring, and Syslog configurations for exporting the log message to the external log management server.

Configure License

Without any license installed on your system, you can evaluate RBFS for 7 days. You need to get an evaluation license or purchase an actual license within 7 days to use the full functionality of RBFS.

The following steps provide the commands to install an RBFS license key. For more information about license configuration, see Installing License.

Switch to config mode using the switch-mode config command to continue with the RBFS configurations.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> switch-mode config
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg>
```

Install the license encrypted string (that is received from RtBrick) using the RBFS CLI.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> set system license <license-
key>
```

RBFS license configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show config system license
AAAAWsfg&jdkfs4D34H5@2evf...
```

As shown below, the "show system license" command displays the expiration date for the current license.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show system license
License Validity:
  License index 1:
    Start date : Tue Feb 28 09:44:27 GMT +0000 2023
    End date   : Mon Mar 04 09:44:27 GMT +0000 2024
supervisor@rtbrick>multiservice-edge.rtbrick.net: op>
```

Configure Instance

Instance `default` will be available by default without any configurations.

Create the instance `inband_mgmt` by entering the following commands.

```
set instance inband_mgmt
set instance inband_mgmt address-family ipv4 unicast
commit
```

The configurations of the instance `inband_mgmt` are shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config instance
inband_mgmt
{
  "rtbrick-config:instance": [
    {
      "name": "inband_mgmt",
      "address-family": [
        {
          "afi": "ipv4",
          "safi": "unicast"
        }
      ]
    }
  ]
}
```

Below are the configurations available for the available instances.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show instance detail
Instance: default
  Instance ID: 0
```

```
  State: Active
  AFI        SAFI             State
  ipv4       unicast          Active
  ipv4       multicast        Active
  ipv4       labeled-unicast  Active
  ipv6       unicast          Active
  ipv6       multicast        Active
  ipv6       labeled-unicast  Active
  mpls       unicast          Active
Instance: inband_mgmt
  Instance ID: 3
  State: Active
  AFI        SAFI             State
  ipv4       unicast          Active
```

Configure Loopback Interface

Loopback Interface configuration is required as it is the best way to identify a network device and is always reachable. Also, protocols use the loopback address to determine protocol-specific properties for the device.

The following steps provide the commands to configure the loopback interface. For more information about Loopback Interface configuration, see Interfaces User Guide.

Configure loopback interface on the device.

```
set interface lo-0/0/1 unit 0 address ipv4 192.0.2.64/32
set interface lo-0/0/1 unit 1 address ipv4 192.0.2.74/32
set interface lo-0/0/1 unit 2 instance inband_mgmt
set interface lo-0/0/1 unit 2 address ipv4 192.0.2.128/32
set interface lo-0/0/1 unit 3 instance inband_mgmt
set interface lo-0/0/1 unit 3 address ipv4 192.0.2.131/32
commit
```



The configuration commands should be followed by the `commit` command to save the configurations into the device.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> commit
```

Loopback Interface configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config interface lo-0/0/1
{
  "rtbrick-config:interface": [
    {
      "name": "lo-0/0/1",
      "unit": [
```

```
        {
          "unit-id": 0,
          "address": {
            "ipv4": [
              {
                "prefix4": "192.0.2.64/32"
              }
            ]
          }
        },
        {
          "unit-id": 1,
          "address": {
            "ipv4": [
              {
                "prefix4": "192.0.2.74/32"
              }
            ]
          }
        },
        {
          "unit-id": 2,
          "instance": "inband_mgmt",
          "address": {
            "ipv4": [
              {
                "prefix4": "192.0.2.128/32"
              }
            ]
          }
        },
        {
          "unit-id": 3,
          "instance": "inband_mgmt",
          "address": {
            "ipv4": [
              {
                "prefix4": "192.0.2.131/32"
              }
            ]
          }
        }
      ]
    }
  ]
}
```

Configure IP Addresses for Core Interfaces

Enter the following commands to configure IP addresses for the core interfaces.

```
set interface ifp-0/1/31 unit 10
set interface ifp-0/1/31 unit 10 vlan 10
set interface ifp-0/1/31 unit 10 address ipv4 192.0.2.1/27
set interface ifp-0/1/31 unit 10 address ipv6 2001:db8::1/64
set interface ifp-0/1/31 unit 100
set interface ifp-0/1/31 unit 100 vlan 100
set interface ifp-0/1/31 unit 100 address ipv4 192.0.2.33/27
set interface ifp-0/1/31 unit 200
set interface ifp-0/1/31 unit 200 instance inband_mgmt
set interface ifp-0/1/31 unit 200 vlan 200
set interface ifp-0/1/31 unit 200 address ipv4 192.0.2.97/27
commit
```

Below configuration shows the IP address configurations for the core interfaces.

```
{
    "rtbrick-config:interface": [
      {
        "name": "ifp-0/1/31",
        "unit": [
          {
            "unit-id": 10,
            "vlan": 10,
            "address": {
              "ipv4": [
                {
                  "prefix4": "192.0.2.1/27"
                }
              ],
              "ipv6": [
                {
                  "prefix6": "2001:db8::1/64"
                }
              ]
            }
          },
          {
            "unit-id": 100,
            "vlan": 100,
            "address": {
              "ipv4": [
                {
                  "prefix4": "192.0.2.33/27"
                }
              ]
            }
          },
          {
            "unit-id": 200,
```

```
            "instance": "inband_mgmt",
            "vlan": 200,
            "address": {
              "ipv4": [
                {
                  "prefix4": "192.0.2.97/27"
                }
              ]
            }
          }
        ]
      }
    ]
  }
}
```

Configure Static Routes to Enable Reachability to the NTP and TACACS Servers

Below are static routes configured to enable reachability to the NTP (192.0.2.129) and TACACS (192.0.2.130) servers. On the Service Node, 192.0.2.98 is the interface address on VLAN 200. It is explained later in this document how to configure IP addresses on Service Node. For details, see Starting BNG Blaster.

```
set instance inband_mgmt static route ipv4 192.0.2.129/32 unicast np1
set instance inband_mgmt static route ipv4 192.0.2.130/32 unicast np1
set instance inband_mgmt static nexthop-profile np1
set instance inband_mgmt static nexthop-profile np1 nexthop 192.0.2.98
set instance inband_mgmt static nexthop-profile np1 lookup-instance inband_mgmt
set instance inband_mgmt static nexthop-profile np1 lookup-afi ipv4
set instance inband_mgmt static nexthop-profile np1 lookup-safi unicast
commit
```

The configuration of the static routes is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config instance
inband_mgmt static
{
  "rtbrick-config:static": {
    "route": {
      "ipv4": [
        {
          "prefix4": "192.0.2.129/32",
          "safi": "unicast",
          "nexthop-profile": "np1"
        },
        {
          "prefix4": "192.0.2.130/32",
          "safi": "unicast",
```

```
          "nexthop-profile": "np1"
        }
      ]
    },
    "nexthop-profile": [
      {
        "name": "np1",
        "nexthop": "192.0.2.98",
        "lookup-instance": "inband_mgmt",
        "lookup-afi": "ipv4",
        "lookup-safi": "unicast"
      }
    ]
  }
}
```

Configure NTP

Configuring NTP (Network Time Protocol) provides time synchronization across a whole network of devices. An NTP network consists devices (clients) which are to be synchronized with the NTP server that provides accurate time to the client devices.

The following steps provide the commands to configure Network Time Protocol (NTP) for the device. For more information about NTP configuration, see NTP User Guide.

Enabling NTP Service:

To access the NTP service running in the RtBrick Host, this service has to be enabled in inband-management. On configuring this, the hosts reachable in inband instance via the physical interface can access this service.

Configure NTP server and NTP service on the device.

```
set system ntp server ntp1
set system ntp server ntp1 ipv4-address 192.0.2.129
set inband-management instance inband_mgmt
set inband-management instance inband_mgmt ntp true
commit
```

NTP configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config inband-management
{
    "rtbrick-config:inband-management": {
      "instance": [
        {
          "name": "inband_mgmt",
          "ntp": "true"
        }
```

```
      ]
    }
  }
```

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config system ntp
{
    "rtbrick-config:ntp": {
      "server": [
        {
          "server-name": "ntp1",
          "ipv4-address": "192.0.2.129"
        }
      ]
    }
  }
```

User Authentication

RBFS supports user authentication through a centralized TACACS+ server and with a local authentication system. The following authentication process typically occurs when a user attempts to access the network.

1. When a user logs in through SSH, the SSH Daemon (sshd) invokes the Pluggable Authentication Module (PAM) to trigger authentication process.
2. PAM requests TACACS+ authentication (except for the user with the supervisor privileges).
3. TACACS+ server provides 'grant access' node if the user authentication is successful.
4. If the user is not allowed using the TACACS+ authentication, it is required to undergo an additional authentication phase. PAM looks up local users. Upon successful authentication, PAM generates RTB PAM token; includes user role in 'scope'.

Define Users on TACACS+ Server

Administrator needs to define users and associate them with the predefined roles on the TACACS+ server. Optionally, RBFS CLI commands can be restricted using the rtb-allow-cmds and rtb-deny-cmds.

The tac_plus.conf file contains configuration information for the tac_plus(tacacs+) daemon. This file is stored at the following location:

/etc/tacacs+/tac_plus.conf

To view the TACACS+ server configuration file, enter the following command.

```
sudo cat /etc/tacacs+/tac_plus.conf
```

For more information about TACACS+ server configuration, see

https://manpages.ubuntu.com/manpages/trusty/man5/tac_plus.conf.5.html

This Reference Design document uses the default local user `supervisor` for the configurations, whereas other users, defined in the TACACS server, can log into RBFS by using their usernames and passwords.

The following TACACS+ configuration shows the details of the TACACS users.

- Click here to download the `tac_plus.conf` file.

Configure TACACS+ on RBFS

After defining the users on the TACACS+ server, configure the TACACS+ server on Multiservice Edge. This configuration allows the remote TACACS+ server to communicate with the Multiservice Edge and to validate user access on the network.

The following steps provide the commands to configure TACACS+. For more information about TACACS+ configuration, see Configure TACACS+ on RBFS.

To access the TACACS+ service running in the RtBrick Host, this service has to be enabled in inband management. On configuring this, the hosts reachable in inband instance via the physical interface can access this service.

```
set system secure-management-status true
set system authorization tacacs 192.0.2.130 inband secret-plain-text
RtBrick_Little_Secret
set inband-management instance inband_mgmt tacacs true
commit
```

In the above configuration, the command `set inband-management instance inband_mgmt tacacs true` is used to enable TACACS+ under the instance called `inband_mgmt`.

TACACS+ configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config system
authorization
{
  "rtbrick-config:authorization": {
    "tacacs": [
      {
        "ipv4-address": "192.0.2.130",
        "type": "inband",
        "secret-encrypted-text":
"$22464b2c7336cfe71e596c447be28d598b9b7b37f92faea157fd5058e5fe0d769"
      }
    ]
  }
}
```

Configuration for enabling TACACS+ under the instance `inband_mgmt` is shown below:

```
"rtbrick-config:inband-management": {
      "instance": [
        {
          "name": "inband_mgmt",
          "tacacs": "true"
        }
      ]
    },
```

Enabling TACACS+ Service on the Service Node

Enter the following commands to enable the TACACS service on the Service Node.

```
~$ sudo /bin/systemctl enable tacacs_plus.service
~$ sudo /bin/systemctl start tacacs_plus.service
```

Validating TACACS+ authentication

The following scenario shows a successful authentication for the user `bob` with password `bob`.

```
~$ ssh bob@multiservice-edge.rtbrick.net
bob@multiservice-edge.rtbrick.net's password:
Last login: Mon Apr 3 16:13:40 2023 from multiservice-edge.rtbrick.net
bob@rtbrick>multiservice-edge.rtbrick.net: op>
```

The following scenario shows an unsuccessful password authentication for the user `bob` with password `bob123`.

```
~$ ssh bob@rtbrick>multiservice-edge.rtbrick.net:
bob@multiservice-edge.rtbrick.net's password:
Permission denied, please try again.
bob@multiservice-edge.rtbrick.net's password:
```

The following scenario shows an unsuccessful authentication for an undefined user `frank`.

```
~$ ssh frank@rtbrick>multiservice-edge.rtbrick.net:
frank@multiservice-edge.rtbrick.net's password:
Permission denied, please try again.
frank@multiservice-edge.rtbrick.net's password:
accounting file = /var/log/tac_plus.acct
key = RtBrick_Little_Secret
```

Configure User Management

Configuring Local User Management enables administrators to create, manage, and secure the users and groups. It allows creation of privileges that are configurable for user-defined and predefined roles.

The following steps provide the commands to configure user management. For more information about license configuration, see Local User Management.

1. To create a role, configure the RBAC privilege and the command privilege. To configure the RBAC privilege for both table and object:

```
set system user admin role supervisor
set system user admin shell /bin/bash
set system user admin password-hashed-text
$6$XNkmuMRI.5.R/NBJ$XDfZec7gEM3z/3lYn8mDDWimRZ/68xawia.pTMdrGqoYHEE3nWHB08DeaPNQTwHW6W
jB1aX6.xjYjh8CNCy4g1
commit
```

For information about Configuring hashed password, see Configure Hashed Password.

The authentication configuration of a password hashed text and an SSH public key is shown below:

```
{
  "ietf-restconf:data": {
    "rtbrick-config:system": {
      "user": [
        {
          "username": "admin",
          "shell": "/usr/local/bin/cli",
          "password-hashed-text":
"$5$L2DaOYYuddhBV$9RA5MX9RQzLC9fIKJzbnoFBb88w9rkSXl7GVrVJ9PY7",
          "ssh-pub-key": [
            "ssh-rsa AAAAWsfg&jdkfs4D34H5@2evf....."
            ]
        }
      ]
    }
  }
}
```

Configure Syslog

RBFS supports sending log messages to a Syslog server. The Syslog configuration can be performed in RBFS.

To configure logging for `bgp` by using Syslog, enter the following commands.

```
set log module bgp
set log module bgp level debug
set log module bgp plugin-alias
set log module bgp plugin-alias alias-name syslog
set log module bgp plugin-alias level debug
commit
```



For event logging, CtrlD only supports Graylog and Syslog. Graylog must be disabled in order to enable Syslog. In addition, Graylog attributes must be replaced with Syslog attributes.

Sylog configuration for the module `bgp` is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config log
{
  "rtbrick-config:log": {
    "module": [
      {
        "module-name": "bgp",
        "level": "debug",
        "plugin-alias": {
          "alias-name": "syslog",
          "level": "debug"
        }
      }
    ]
  }
}
```

Accessing the RtBrick Host to Configure Syslog

The steps described in this section are performed on the RtBrick Host. For logging into the RtBrick Host, use SSH port 1022.

```
ssh supervisor@<multiservice-edge-management-ip> -p 1022
```

After logging into the RtBrick Host, go to the following location of CtrlD and edit the `config.json` file.

- `/etc/rtbrick/ctrld/config.json`

Specify the Syslog configurations as shown below in the `config.json` file.

```
{
    "rbms_enable": false,
    "graylog_enable": false,
    "syslog_enable": true,
```

```
    "syslog_network": "udp",
    "syslog_urls": [
      "198.51.100.49:516"
    ],
    "syslog_severity_level": 7,
    "auth_enabled": false
}
```



- For documentation purposes, the IP address `198.51.100.49` has been used as the IP address of the Syslog endpoint. This IP address should be updated with the actual Syslog server's IP address.
- Syslog messages can be transported using UDP or TCP protocol. In this configuration, Syslog messages are transported using `udp`.

After making configuration changes in the `config.json`, restart CtrlD service as shown below.

```
supervisor@onl>multiservice-edge.rtbrick.net:~ $ sudo service rtbrick-ctrld restart
[sudo] password for supervisor:
[ ok ] Stopping rtbrick ctrld service:.
[ ok ] Starting rtbrick ctrld service:.
```

Monitor Resources (Resmon)

Resource monitoring enables administrators to collect and analyze the health information and usage data of various hardware resources such as CPU, memory, processes, disks, sensors, optics, and so on.

Run `show cpu usage`, `show memory usage` and `show disk usage` to see the CPU, memory and disk utilization respectively.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show cpu usage
Name          Total       User     System       Nice   I/O Wait       Idle        IRQ
Soft IRQ
cpu             11%         9%         1%         0%         0%        88%         0%
      0%
cpu0            10%         8%         1%         0%         0%        89%         0%
      0%
cpu1            44%        43%         1%         0%         0%        55%         0%
      0%
cpu2            34%        32%         2%         0%         0%        66%         0%
      0%
cpu3            11%         9%         2%         0%         0%        89%         0%
      0%
cpu4             3%         1%         2%         0%         0%        96%         0%
      0%
cpu5             4%         3%         1%         0%         0%        96%         0%
      0%
```

```
cpu6            13%         11%          2%          0%          0%         87%          0%
      0%
cpu7            24%         22%          1%          0%          0%         75%          0%
      0%
cpu8             4%          2%          2%          0%          0%         95%          0%
      0%
cpu9             2%          1%          1%          0%          0%         97%          0%
      0%
cpu10            6%          4%          2%          0%          0%         93%          0%
      0%
cpu11            3%          2%          1%          0%          0%         96%          0%
      0%
cpu12            8%          6%          2%          0%          0%         91%          0%
      0%
cpu13            3%          3%          0%          0%          0%         96%          0%
      0%
cpu14            6%          4%          2%          0%          0%         93%          0%
      0%
cpu15            8%          5%          2%          0%          0%         91%          0%
      0%
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show memory usage
Name     Total            Used             Free          Shared       Buffers
 Cached
RAM      31.03 GiB        8.51 GiB         17.04 GiB    1.19 GiB     112.66 MiB   5.37
GiB
SWAP     0 bytes          0 bytes          0 bytes      n/a          n/a          n/a
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show disk usage
Filesystem               Type             Size         Used         Available
 Mountpoint          Usage %
none                     tmpfs            492 KiB      0 bytes      492 KiB      /dev
               0.0
tmpfs                    tmpfs            15.51 GiB    17.23 MiB    15.5 GiB     /run
               0.11
tmpfs                    tmpfs            6 GiB        828.75 MiB   5.19 GiB     /shm
               13.49
tmpfs                    tmpfs            15.51 GiB    182.96 MiB   15.33 GiB
 /dev/shm            1.15
tmpfs                    tmpfs            5 MiB        0 bytes      5 MiB
 /run/lock           0.0
devtmpfs                 devtmpfs         1 MiB        0 bytes      1 MiB
 /dev/mem            0.0
/dev/sda10               ext4             15.62 GiB    50.61 MiB    14.76 GiB
 /var/log            0.33
/dev/sda6                ext4             29.4 GiB     4.24 GiB     23.65 GiB
 /platform           15.2
tmpfs                    tmpfs            3.1 GiB      0 bytes      3.1 GiB
 /run/user/1000      0.0
tmpfs                    tmpfs            3.1 GiB      0 bytes      3.1 GiB
 /run/user/1001      0.0
tmpfs                    tmpfs            15.51 GiB    0 bytes      15.51 GiB
 /sys/fs/cgroup      0.0
```

```
/dev/sda11               ext4            43.79 GiB    51.89 MiB    41.49 GiB
 /var/crash          0.12
tmpfs                    tmpfs           3.1 GiB      1.02 MiB     3.1 GiB
 /var/run-ext/onl/r  0.03
/var/cache/rtbrick/imag  overlay         29.4 GiB     4.24 GiB     23.65 GiB    /
              15.2
```

The show command can also be used to view other resource details. For information about the resmon configuration and operational commands, see the RBFS Resource Monitoring Guide.

3.3. Protocol Configurations

This validated solution design topology uses OSPFv2/v3 as the interior gateway protocols to distribute IP routing information among the routers in an Autonomous System (AS). The Label Distribution Protocol (LDP) is used to exchange label mapping information for MPLS traffic. And, iBGP is used for exchanging routing and reachability information within ASs.

One thus needs to configure the following protocols:

- OSPFv2/v3: To ensure IP connectivity on the core network.
- LDP: To establish MPLS LSP tunnels for MPLS data transmission on the network.
- iBGP: To exchange routing information within an AS.

3.3.1. Configure OSPFv2/v3

The following steps provide the commands to execute various functionalities of the OSPFv2/v3 protocols. For more detailed information about OSPF configuration, see OSPF User Guide.

1. Configure OSPFv2/v3 address family, hostname, interface, area, segment routing, and instance ID (for IPv6).

```
set instance default protocol ospf address-family ipv4 hostname multiservice-edge
set instance default protocol ospf address-family ipv4 segment-routing srgb base
100000
set instance default protocol ospf address-family ipv4 segment-routing srgb range 2000
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-
0/1/17/10
set instance default protocol ospf address-family ipv4 area 0.0.0.0 interface ifl-
0/1/17/10 network-type p2p
set instance default protocol ospf address-family ipv4 area 0.0.0.0 interface lo-
0/0/1/0
set instance default protocol ospf address-family ipv4 area 0.0.0.0 interface lo-
0/0/1/0 segment-routing index 100
set instance default protocol ospf address-family ipv6 instance-id 0
set instance default protocol ospf address-family ipv6 instance-id 0 hostname
multiservice-edge
set instance default protocol ospf address-family ipv6 instance-id 0 segment-routing
```

```
srgb base 200000
set instance default protocol ospf address-family ipv6 instance-id 0 segment-routing
srgb range 2000
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
interface ifl-0/1/17/10
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
interface ifl-0/1/17/10 network-type p2p
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
interface lo-0/0/1/0
set instance default protocol ospf address-family ipv6 instance-id 0 area 0.0.0.0
interface lo-0/0/1/0 segment-routing index 100
commit
```

OSPFv2/v3 instance configuration on interface is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config instance default
protocol ospf address-family
{
  "rtbrick-config:address-family": {
    "ipv4": {
      "hostname": "multiservice-edge",
      "segment-routing": {
        "srgb": {
          "base": 100000,
          "range": 2000
        }
      },
      "area": [
        {
          "area-id": "0.0.0.0",
          "interface": [
            {
              "name": "ifl-0/1/17/10",
              "network-type": "p2p"
            },
            {
              "name": "lo-0/0/1/0",
              "segment-routing": {
                "index": 100
              }
            }
          ]
        }
      ]
    },
    "ipv6": {
      "instance-id": [
        {
          "instance-id": 0,
```

```
          "hostname": "multiservice-edge",
          "segment-routing": {
            "srgb": {
              "base": 200000,
              "range": 2000
            }
          },
          "area": [
            {
              "area-id": "0.0.0.0",
              "interface": [
                {
                  "name": "ifl-0/1/17/10",
                  "network-type": "p2p"
                },
                {
                  "name": "lo-0/0/1/0",
                  "segment-routing": {
                    "index": 100
                  }
                }
              ]
            }
          ]
        }
      ]
    }
  }
}
```

3.3.2. Configure LDP on the Interfaces

The following steps provide the commands to execute various LDP functionalities. For more detailed information about LDP configuration, see *LDP User Guide*.

1. Configure LDP on the router interface.

```
set instance default protocol ldp router-id 192.0.2.64
set instance default protocol ldp interface ifl-0/0/1/10
set instance default protocol ldp interface lo-0/0/1/0
commit
```

Configuration for LDP on the interface is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config instance default protocol ldp
{
  "rtbrick-config:ldp": {
    "router-id": "192.0.2.64",
    "interface": [
```

```
      {
        "name": "ifl-0/0/1/10"
      },
      {
        "name": "lo-0/0/1/0"
      }
    ]
  }
}
```

3.3.3. Configure BGP

The following steps provide the commands to execute the various BGP functionalities quickly. For more detailed information about BGP configuration, see BGP User Guide.

1. Configure BGP local AS, router-id, and hostname

```
set instance default protocol bgp local-as 4200000001
set instance default protocol bgp router-id 192.0.2.64
commit
```

BGP local AS, router-id, and hostname configurations are shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config instance default protocol bgp
{
  "rtbrick-config:bgp": {
    "local-as": 4200000001,
    "router-id": "192.0.2.64",

<....>
```

2. Enable the IPv4 and IPv6 address families which are to be supported on the specific BGP instance.

```
set instance default protocol bgp address-family ipv4 unicast
set instance default protocol bgp address-family ipv4 unicast resolve-nexthop safi labeled-unicast
set instance default protocol bgp address-family ipv4 unicast redistribute direct
set instance default protocol bgp address-family ipv6 labeled-unicast
set instance default protocol bgp address-family ipv6 unicast
set instance default protocol bgp address-family ipv6 unicast resolve-nexthop safi labeled-unicast
set instance default protocol bgp address-family ipv6 unicast redistribute direct
commit
```

BGP address family configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config instance default protocol bgp address-family
{
  "rtbrick-config:address-family": [
    {
```

```
      "afi": "ipv4",
      "safi": "unicast",
      "resolve-nexthop": {
        "safi": "labeled-unicast"
      },
      "redistribute": [
        {
          "source": "direct"
        }
      ]
    },
    {
      "afi": "ipv6",
      "safi": "labeled-unicast"
    },
    {
      "afi": "ipv6",
      "safi": "unicast",
      "resolve-nexthop": {
        "safi": "labeled-unicast"
      },
      "redistribute": [
        {
          "source": "direct"
        }
      ]
    }
  ]
}
```

3. Create the peer group with the specific remote AS configurations and the address family that is to be negotiated with the peer which will be attached to the peer group later.

```
set instance default protocol bgp peer-group RR
set instance default protocol bgp peer-group RR remote-as 4200000001
set instance default protocol bgp peer-group RR address-family ipv4 unicast
set instance default protocol bgp peer-group RR address-family ipv6 labeled-unicast
set instance default protocol bgp peer-group RR address-family ipv6 unicast
set instance default protocol bgp peer-group RR address-family ipv6 unicast send-label true
commit
```

BGP peer-group configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config instance default protocol bgp peer-group
{
  "rtbrick-config:peer-group": [
    {
      "pg-name": "RR",
      "remote-as": 4200000001,
      "address-family": [
        {
          "afi": "ipv4",
          "safi": "unicast"
        },
        {
          "afi": "ipv6",
```

```
          "safi": "labeled-unicast"
        },
        {
          "afi": "ipv6",
          "safi": "unicast",
          "send-label": "true"
        }
      ]
    }
  ]
}
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg>
```

4. Add a BGP peer and associate it with the specific peer group.

```
set instance default protocol bgp peer
set instance default protocol bgp peer ipv4 198.51.100.3 192.0.2.64
set instance default protocol bgp peer ipv4 198.51.100.3 192.0.2.64 peer-group RR
commit
```

Configuration for adding a BGP peer and associating it with a peer group is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config instance default protocol bgp peer
{
  "rtbrick-config:peer": {
    "ipv4": [
      {
        "peer-address": "198.51.100.3",
        "update-source": "192.0.2.64",
        "peer-group": "RR"
      }
    ]
  }
}
```

5. Configure the IPv6 unicast address family with `send-label` as true, then address-family IPv6 labeled-unicast gets negotiated with the peer.

```
set instance default protocol bgp peer-group RR address-family ipv6 unicast send-label true
commit
```

The following configuration shows the BGP IPv6 unicast address family with `send-label` as true.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config instance default protocol bgp peer-group RR address-
family ipv6 unicast
{
  "rtbrick-config:address-family": [
    {
      "afi": "ipv6",
      "safi": "unicast",
      "send-label": "true"
    }
```

```
  ]
}
```

6. Set the resolve-nexthop, if the BGP nexthop attribute of the BGP routes needs to be resolved under ipv4/ipv6 labeled-unicast routing table. It configures only resolve-nexthop safi. Based on the nexthop-type (ipv4 or ipv6), it gets looked up into either IPv4 labeled-unicast or IPv6 labeled-unicast.

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

Resolve nexthop configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config instance default protocol bgp address-family ipv4
unicast resolve-nexthop
{
  "rtbrick-config:resolve-nexthop": {
    "safi": "labeled-unicast"
  }
}
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg>
```

7. To redistribute the routes (belonging to a specific source) into BGP, execute the following command. The following command redistributes `direct` routes into BGP.

```
set instance default protocol bgp address-family ipv4 unicast redistribute direct
commit
```

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config instance default protocol bgp address-family ipv4
unicast redistribute
{
  "rtbrick-config:redistribute": [
    {
      "source": "direct"
    }
  ]
}
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg>
```

3.3.4. Configuring the Service Node

Configuring Interfaces on the Service Node for RADIUS Connectivity from Multiservice Edge

To configure interfaces on the Service Node for RADIUS connectivity from Multiservice Edge, enter the following commands:

```
sudo ip link add link SN-2-DT name SN-2-DT.100 type vlan id 100
sudo ifconfig SN-2-DT.100 192.0.2.34/27
```



SN-2-DT is the internal nomenclature that denotes the interface name on Service Node that connects to the Multiservice Edge.

Configuring Interfaces on the Service Node for NTP and TACACS Connectivity from Multiservice Edge

To configure interfaces on the Service Node for NTP and TACACS connectivity from Multiservice Edge, enter the following commands:

```
sudo ip link add link SN-2-DT name SN-2-DT.200 type vlan id 200
sudo ifconfig SN-2-DT.200 192.0.2.98/27
sudo ifconfig lo:1 192.0.2.129 netmask 255.255.255.255 up
sudo ifconfig lo:2 192.0.2.130 netmask 255.255.255.255 up
```

Configuring Routes on the Service Node

To configure routes on the Service Node that provide reachability to the RADIUS, TACACS and NTP servers, enter the following commands:

```
sudo ip route add 192.0.2.74/32 via 192.0.2.33
sudo ip route add 192.0.2.131/32 via 192.0.2.97
sudo ip route add 192.0.2.128/32 via 192.0.2.97
```

BNG Blaster Configuration for Protocols

BNG Blaster is an open-source network testing platform for access and routing protocols. It can emulate massive PPPoE and IPoE (DHCP) subscribers including IPTV, and L2TP (LNS). There are various routing protocols supported such as OSPFv2/v3 and BGP. So, one can use this platform for end-to-end BNG and non-BNG router testing.

For more information about BNG Blaster, see https://github.com/rtbrick/bngblaster

For information about installing BNG Blaster, see https://rtbrick.github.io/bngblaster/install.html

Downloading the Blaster Configuration File

The following is the configuration file that is used in BNG Blaster for validating PPPoE, IPoE, BGP, OSPFv2/v3, and LDP.

- Click here to download the BNG Blaster configuration file (`blaster_cgnat.json`).

Generating Supporting Files for Protocols

- **Generating BGP Internet Prefixes**

Enter the following commands to generate BGP internet prefixes.

```
bgpupdate -f internet.bgp -a 4200000001 -n 198.51.100.2 -N 1 -p 172.16.0.0/24 -P 1000000
```

```
bgpupdate -f internet.bgp -a 4200000001 -n 198.51.100.2 -N 1 -p 2004::/48 -m 10000 -M 5 -P 150000 --append
```

Ensure that the command execution has finished (as shown below) before continuing.

```
[2023-04-05 10:19:32][INFO   ] init 1000000 IPv4 prefixes
[2023-04-05 10:19:56][INFO   ] open file internet.bgp (replace)
[2023-04-05 10:25:31][INFO   ] finished
```

After the generation of the `internet.bgp` file, the "raw-update-file" attribute of the `blaster_cgnat.json` file needs to be updated as follows:

```
"raw-update-file": "/home/supervisor/internet.bgp"
```



For more information about downloading the blaster_cgnat.json file, see section Downloading the Blaster Configuration File.

- **Generating MRT File for OSPF**

The `.mrt` file, which is used by BNG Blaster, represents the simulated topology of the network that includes R-1 and R-2.

Below are the JSON files (`ospfv2_3node.json` and `ospfv3_3node.json`) which are used to simulate R-1, R-2, and RR on BNG Blaster.

- Click here to download the `ospfv2_3node.json` file.
- Click here to download the `ospfv3_3node.json` file.

The `ospfv2_3node.json` file needs to be converted to MRT format using the following command:

```
lspgen -r ospfv2_3node.json -m ospf.mrt
```

The `ospfv3_3node.json` file needs to be converted to MRT format using the following command:

```
lspgen -r ospfv3_3node.json -m ospfv3.mrt
```

After converting the files to `ospf.mrt` and `ospfv3.mrt`, it needs to be updated in the OSPF section in the `blaster_cgnat.json` file as shown below.

```
"mrt-file": "/home/supervisor/ospfv3.mrt"
```


```
"mrt-file": "/home/supervisor/ospfv3.mrt"
```

- Generate labels for the OSPF prefixes using "ldpupdate" command as shown below:

```
ldpupdate -l 192.0.2.65 -p 198.51.100.1/32 -P 3 -M 3 -f out.ldp
```

The detail of the generated file needs to be added to the LDP section in the `blaster_cgnat.json` file as shown below:

```
"raw-update-file": "/home/supervisor/out.ldp"
```

Starting BNG Blaster

In the following command line string, a BNG Blaster instance is started and the `blaster_cgnat.json` file is used.

```
sudo bngblaster -C blaster_cgnat.json -I
```

The **-C blaster_cgnat.json** argument specifies the blaster configuration file. The -I flag enables the interactive blaster UI.



Figure 12. Starting BNG Blaster

3.3.5. Validating Protocols on RBFS Multiservice Edge

Validating OSPF Adjacency, Routes and Reachability

Run the following command to show OSPF adjacency.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show ospf neighbor
Instance: default, Address family: ipv4
  Address                       Interface           Router          Area            State      Priority  DR
   BDR            Uptime          Expires
  192.0.2.2                     ifl-0/0/1/10      R1              0.0.0.0         Full       64        0.0.0.0
0.0.0.0         0d:00h:00m:03s  38s
```

After configuring OSPF protocol, check the IPv4 unicast routes, populated by OSPF using the following command:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show route ipv4 unicast source ospf instance default
Instance: default, AFI: ipv4, SAFI: unicast
Prefix/Label                            Source           Pref    Next Hop                                  Interface
192.0.2.65/32                           ospf             10      192.0.2.2                                 ifl-0/0/1/10
198.51.100.1/32                         ospf             10      192.0.2.2                                 ifl-0/0/1/10
198.51.100.2/32                         ospf             10      192.0.2.2                                 ifl-0/0/1/10
198.51.100.3/32                         ospf             10      192.0.2.2                                 ifl-0/0/1/10
198.51.100.100/31                       ospf             10      192.0.2.2                                 ifl-0/0/1/10
198.51.100.102/31                       ospf             10      192.0.2.2                                 ifl-0/0/1/10
198.51.100.104/31                       ospf             10      192.0.2.2                                 ifl-0/0/1/10
```

Ping the address 192.0.2.65 as follows:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> ping 192.0.2.65
68 bytes from 192.0.2.65: icmp_seq=1 ttl=64 time=8.2474 ms
68 bytes from 192.0.2.65: icmp_seq=2 ttl=64 time=1.3832 ms
68 bytes from 192.0.2.65: icmp_seq=3 ttl=64 time=4.2682 ms
68 bytes from 192.0.2.65: icmp_seq=4 ttl=64 time=1.1797 ms
68 bytes from 192.0.2.65: icmp_seq=5 ttl=64 time=1.2778 ms
Statistics: 5 sent, 5 received, 0% packet loss
```

Validating LDP Adjacency, Routes and Reachability

Run the following commands to show LDP neighbor and LDP session.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show ldp neighbor
Instance: default
  Interface/Peer      LDP ID              Transport IP             Up Since                  Expires in 14s
  ifl-0/0/1/10     192.0.2.65:0        192.0.2.65               Thu Mar 21 06:27:38
```

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show ldp session
Instance: default
  LDP ID            Peer IP                  State             Up/Down           FECRcvd   FECSent
  192.0.2.65:0      192.0.2.65               Operational       0d:00h:02m:56s          5         5
```

After configuring the LDP protocol, check the IPv4 labeled unicast routes, populated by LDP using the following command:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show  route ipv4 labeled-unicast source ldp
Instance: default, AFI: ipv4, SAFI: labeled-unicast
```

```
Prefix/Label                            Source          Pref    Next Hop
Interface                   Label
192.0.2.65/32                           ldp             9       192.0.2.2                               ifl-
0/0/1/10                -
198.51.100.1/32                         ldp             9       192.0.2.2                               ifl-
0/0/1/10                10000
198.51.100.2/32                         ldp             9       192.0.2.2                               ifl-
0/0/1/10                10001
198.51.100.3/32                         ldp             9       192.0.2.2                               ifl-
0/0/1/10                10002
```

Ping the labeled unicast address 198.51.100.1 as follows:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> ping 198.51.100.1 instance default afi ipv4 safi labeled-unicas
t
68 bytes from 198.51.100.1: icmp_seq=1 ttl=64 time=.2344 ms
68 bytes from 198.51.100.1: icmp_seq=2 ttl=64 time=4.1493 ms
68 bytes from 198.51.100.1: icmp_seq=3 ttl=64 time=6.9031 ms
68 bytes from 198.51.100.1: icmp_seq=4 ttl=64 time=.2507 ms
68 bytes from 198.51.100.1: icmp_seq=5 ttl=64 time=3.2822 ms
Statistics: 5 sent, 5 received, 0% packet loss
```



The command argument `labeled-unicast` takes the ICMP requests through a labeled path while validating IP connectivity and hence, it prepends an MPLS label.

Validating BGP Adjacency, Routes and Reachability

Run the following commands to show BGP session and state.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show bgp peer
Instance: default
  Peer                                  Remote AS    State         Up/Down Time              PfxRcvd
PfxSent
  198.51.100.3                          4200000001   Established   0d:00h:04m:14s            1150000             5
```

After configuring BGP, check the IPv4 unicast routes, populated by BGP using the following command:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show route ipv4 unicast source bgp instance default show route ipv4
unicast source bgp instance default
Instance: default, AFI: ipv4, SAFI: unicast
Prefix/Label                            Source          Pref    Next Hop
Interface
172.16.0.0/24                           bgp             200     198.51.100.2                            ifl-
0/0/1/10
172.16.1.0/24                           bgp             200     198.51.100.2                            ifl-
0/0/1/10
172.16.2.0/24                           bgp             200     198.51.100.2                            ifl-
0/0/1/10
172.16.3.0/24                           bgp             200     198.51.100.2                            ifl-
0/0/1/10
172.16.4.0/24                           bgp             200     198.51.100.2                            ifl-
0/0/1/10
172.16.5.0/24                           bgp             200     198.51.100.2                            ifl-
0/0/1/10
```

```
172.16.6.0/24                           bgp             200     198.51.100.2                            ifl-
0/0/1/10
172.16.7.0/24                           bgp             200     198.51.100.2                            ifl-
0/0/1/10
172.16.8.0/24                           bgp             200     198.51.100.2                            ifl-
0/0/1/10
172.16.9.0/24                           bgp             200     198.51.100.2                            ifl-
0/0/1/10
<...>
```

This command will list all the 1 million IPv4 BGP internet prefixes.

Pinging an IPv4 route (source: bgp) from the Multiservice Edge.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> ping 172.16.1.0
68 bytes from 172.16.1.0: icmp_seq=1 ttl=64 time=4.3976 ms
68 bytes from 172.16.1.0: icmp_seq=2 ttl=64 time=1.5301 ms
68 bytes from 172.16.1.0: icmp_seq=3 ttl=64 time=3.0536 ms
68 bytes from 172.16.1.0: icmp_seq=4 ttl=64 time=5.3049 ms
68 bytes from 172.16.1.0: icmp_seq=5 ttl=64 time=4.4273 ms
Statistics: 5 sent, 5 received, 0% packet loss
```

Check the IPv6 unicast routes, populated by BGP using the following command:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show route ipv6 unicast source bgp instance default show route ipv6
unicast source bgp instance default
Instance: default, AFI: ipv6, SAFI: unicast
Prefix/Label                            Source          Pref    Next Hop
Interface
2004::/48                               bgp             200     198.51.100.2                            ifl-
0/0/1/10
2004:0:1::/48                           bgp             200     198.51.100.2                            ifl-
0/0/1/10
2004:0:2::/48                           bgp             200     198.51.100.2                            ifl-
0/0/1/10
2004:0:3::/48                           bgp             200     198.51.100.2                            ifl-
0/0/1/10
2004:0:4::/48                           bgp             200     198.51.100.2                            ifl-
0/0/1/10
2004:0:5::/48                           bgp             200     198.51.100.2                            ifl-
0/0/1/10
2004:0:6::/48                           bgp             200     198.51.100.2                            ifl-
0/0/1/10
2004:0:7::/48                           bgp             200     198.51.100.2                            ifl-
0/0/1/10
2004:0:8::/48                           bgp             200     198.51.100.2                            ifl-
0/0/1/10
2004:0:9::/48                           bgp             200     198.51.100.2                            ifl-
0/0/1/10
2004:0:a::/48                           bgp             200     198.51.100.2                            ifl-
0/0/1/10

<...>
```

Pinging an IPv6 route (source: bgp) from the Multiservice Edge.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> ping 2004:0:1::
68 bytes from 2004:0:1::: icmp_seq=1 ttl=255 time=2.2185 ms
68 bytes from 2004:0:1::: icmp_seq=2 ttl=255 time=1.1854 ms
68 bytes from 2004:0:1::: icmp_seq=3 ttl=255 time=5.3917 ms
68 bytes from 2004:0:1::: icmp_seq=4 ttl=255 time=2.3068 ms
68 bytes from 2004:0:1::: icmp_seq=5 ttl=255 time=5.4419 ms
Statistics: 5 sent, 5 received, 0% packet loss
```

3.4. RBFS Subscriber Management Configuration

This chapter provides information about the following:

- PPPoE Subscriber Management Configuration
- IPoE Subscriber Management Configuration
- QoS Configuration for PPPoE and IPoE Subscribers

3.4.1. PPPoE Subscriber Management Configuration

PPPoE (Point-to-Point Protocol over Ethernet) is a network protocol that provides connectivity across Ethernet networks. The protocol enables communication between network endpoints and service providers implement PPPoE to connect many hosts on a single Ethernet LAN to the core network through a common device. In RBFS, the PPPoE daemon (`pppoed`) manages PPPoE and PPP sessions.

To enable PPPoE Subscriber Management, the following configurations are mandatory:

1. Access Interface Configuration
2. Access Profile Configuration
3. AAA (Authentication, Authorization and Accounting) Profile Configuration. Based on the authentication requirement, configure any one of the following:
    a. Local Authentication
        i. Pool Configuration
        ii. User Profile Configuration
    b. RADIUS Authentication
        i. RADIUS Profile Configuration
        ii. RADIUS Server Configuration

This solution section discusses RADIUS authentication.

NOTES:

- Access interfaces can be configured without VLAN tags (untagged) and with one (single tagged) or two (double tagged) VLAN tags.
- There can be more than one interface configured for subscriber management and each

interface can reference the same profile.

Configuring PPPoE Subscriber Management

For detailed information about the subscriber configuration options, see the Subscriber Management Configuration Guide.

1. Configure the access profile `pppoe`.

```
set access access-profile pppoe
set access access-profile pppoe protocol pppoe enable true
set access access-profile pppoe protocol ppp lcp authentication-protocol PAP
set access access-profile pppoe protocol ppp lcp echo-interval 30
set access access-profile pppoe protocol ppp lcp echo-enable true
set access access-profile pppoe protocol ppp ipcp enable true
set access access-profile pppoe protocol ppp ipcp source-ifl lo-0/0/1/0
set access access-profile pppoe protocol ppp ip6cp enable true
set access access-profile pppoe protocol ra enable true
set access access-profile pppoe protocol ra interval 30
set access access-profile pppoe protocol dhcpv6 enable true
set access access-profile pppoe protocol dhcpv6 lifetime 14400
set access access-profile pppoe protocol dhcpv6 preferred-lifetime 1800
set access access-profile pppoe address-family ipv4 enable true
set access access-profile pppoe address-family ipv4 pool-name poolv4
set access access-profile pppoe address-family ipv4 primary-dns 203.0.113.200
set access access-profile pppoe address-family ipv4 secondary-dns 203.0.113.201
set access access-profile pppoe address-family ipv4 instance default
set access access-profile pppoe address-family ipv6 enable true
set access access-profile pppoe address-family ipv6 pool-name poolv6
set access access-profile pppoe address-family ipv6 prefix-delegation-pool-name
poolv6pd
set access access-profile pppoe address-family ipv6 primary-dns 2001:db8:0:20::1
set access access-profile pppoe address-family ipv6 secondary-dns 2001:db8:0:20::1
set access access-profile pppoe address-family ipv6 instance default
commit
```

The access profile configuration is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config access access-
profile
{
  "rtbrick-config:access-profile": [
    {
      "profile-name": "pppoe",
      "protocol": {
        "pppoe": {
          "enable": "true"
        },
        "ppp": {
```

```
          "lcp": {
            "authentication-protocol": "PAP",
            "echo-interval": 30,
            "echo-enable": "true"
          },
          "ipcp": {
            "enable": "true",
            "source-ifl": "lo-0/0/1/0"
          },
          "ip6cp": {
            "enable": "true"
          }
        },
        "ra": {
          "enable": "true",
          "interval": 30
        },
        "dhcpv6": {
          "enable": "true",
          "lifetime": 14400,
          "preferred-lifetime": 1800
        }
      },
      "address-family": {
        "ipv4": {
          "enable": "true",
          "pool-name": "poolv4",
          "primary-dns": "203.0.113.200",
          "secondary-dns": "203.0.113.201",
          "instance": "default"
        },
        "ipv6": {
          "enable": "true",
          "pool-name": "poolv6",
          "prefix-delegation-pool-name": "poolv6pd",
          "primary-dns": "2001:db8:0:20::1",
          "secondary-dns": "2001:db8:0:20::1",
          "instance": "default"
        }
      }
    }
  ]
}
```

2. Configure the Authentication and Accounting (AAA) profile for `aaa-profile`.

```
set access aaa-profile aaa-profile
set access aaa-profile aaa-profile session-timeout 0
set access aaa-profile aaa-profile idle-timeout 0
set access aaa-profile aaa-profile aaa-radius-profile radius-profile
```

```
set access aaa-profile aaa-profile authentication order RADIUS
set access aaa-profile aaa-profile accounting order RADIUS
set access aaa-profile aaa-profile accounting interim-interval 86400
set access aaa-profile aaa-profile accounting ingress accounting-source POLICER
set access aaa-profile aaa-profile accounting egress accounting-source CLASS
commit
```

The access AAA configuration is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config access aaa-profile
{
  "rtbrick-config:aaa-profile": [
    {
      "profile-name": "aaa-profile",
      "session-timeout": 0,
      "idle-timeout": 0,
      "aaa-radius-profile": "radius-profile",
      "authentication": {
        "order": "RADIUS"
      },
      "accounting": {
        "order": "RADIUS",
        "interim-interval": 86400,
        "ingress": {
          "accounting-source": "POLICER"
        },
        "egress": {
          "accounting-source": "CLASS"
        }
      }
    }
  ]
}
```

3. Configure the access interface. Double-tagged interface is configured in this case as the access interface (`ifp-0/1/3`). The interface configuration assigns the access type, access profile, AAA profile, and further optional attributes like service-profile to the specified access interface.

Commands for PPPoE:

```
set access interface double-tagged ifp-0/1/40 4020 4020 1001 2000
set access interface double-tagged ifp-0/1/40 4020 4020 1001 2000 access-type PPPoE
set access interface double-tagged ifp-0/1/40 4020 4020 1001 2000 access-profile-name
pppoe
set access interface double-tagged ifp-0/1/40 4020 4020 1001 2000 service-profile-name
subs_service
set access interface double-tagged ifp-0/1/40 4020 4020 1001 2000 aaa-profile-name
aaa-profile
set access interface double-tagged ifp-0/1/42 4020 4020 1001 2000
```

```
set access interface double-tagged ifp-0/1/42 4020 4020 1001 2000 access-type PPPoE
set access interface double-tagged ifp-0/1/42 4020 4020 1001 2000 access-profile-name
pppoe
set access interface double-tagged ifp-0/1/42 4020 4020 1001 2000 service-profile-name
subs_service
set access interface double-tagged ifp-0/1/42 4020 4020 1001 2000 aaa-profile-name
aaa-profile
commit
```

The double-tagged access interface configuration (for PPPoE) is shown below.

```
supervisor@rtbrick>ufi08.q2c.u23.r4.nbg.rtbrick.net: cfg> show config access interface
double-tagged
{
  "rtbrick-config:double-tagged": [
    {
      "interface-name": "ifp-0/1/40",
      "outer-vlan-min": 4020,
      "outer-vlan-max": 4020,
      "inner-vlan-min": 1001,
      "inner-vlan-max": 2000,
      "access-type": "PPPoE",
      "access-profile-name": "pppoe",
      "service-profile-name": "subs_service",
      "aaa-profile-name": "aaa-profile"
    },
    {
      "interface-name": "ifp-0/1/42",
      "outer-vlan-min": 4020,
      "outer-vlan-max": 4020,
      "inner-vlan-min": 1001,
      "inner-vlan-max": 2000,
      "access-type": "PPPoE",
      "access-profile-name": "pppoe",
      "service-profile-name": "subs_service",
      "aaa-profile-name": "aaa-profile"
    }
  ]
}
```

Commands for IPoE:

```
set access interface single-tagged ifp-0/1/40 10 4010
set access interface single-tagged ifp-0/1/40 10 4010 access-type IPoE
set access interface single-tagged ifp-0/1/40 10 4010 access-profile-name ipoe
set access interface single-tagged ifp-0/1/40 10 4010 service-profile-name
subs_service
set access interface single-tagged ifp-0/1/40 10 4010 aaa-profile-name aaa-profile
set access interface single-tagged ifp-0/1/40 10 4010 gateway-ifl lo-0/0/1/0
```

```
set access interface single-tagged ifp-0/1/42 10 4010
set access interface single-tagged ifp-0/1/42 10 4010 access-type IPoE
set access interface single-tagged ifp-0/1/42 10 4010 access-profile-name ipoe
set access interface single-tagged ifp-0/1/42 10 4010 service-profile-name
subs_service
set access interface single-tagged ifp-0/1/42 10 4010 aaa-profile-name aaa-profile
set access interface single-tagged ifp-0/1/42 10 4010 gateway-ifl lo-0/0/1/0
commit
```

The single-tagged access interface configuration (for IPoE) is shown below.

```
supervisor@rtbrick>ufi08.q2c.u23.r4.nbg.rtbrick.net: cfg> show config access interface
single-tagged
{
  "rtbrick-config:single-tagged": [
    {
      "interface-name": "ifp-0/1/40",
      "outer-vlan-min": 10,
      "outer-vlan-max": 4010,
      "access-type": "IPoE",
      "access-profile-name": "ipoe",
      "service-profile-name": "subs_service",
      "aaa-profile-name": "aaa-profile",
      "gateway-ifl": "lo-0/0/1/0"
    },
    {
      "interface-name": "ifp-0/1/42",
      "outer-vlan-min": 10,
      "outer-vlan-max": 4010,
      "access-type": "IPoE",
      "access-profile-name": "ipoe",
      "service-profile-name": "subs_service",
      "aaa-profile-name": "aaa-profile",
      "gateway-ifl": "lo-0/0/1/0"
    }
  ]
}
```

4. In this solution, we configure AAA authentication and accounting with RADIUS. To use RADIUS authentication and accounting both the RADIUS profile and RADIUS server configurations (see below) must be configured.

a. Configure RADIUS profile `radius-profile`.

```
set access radius-profile radius-profile
set access radius-profile radius-profile nas-ip-address 192.0.2.74
set access radius-profile radius-profile nas-port-format DEFAULT
set access radius-profile radius-profile nas-port-type Ethernet
set access radius-profile radius-profile authentication radius-server-profile-name
```

```
radius
set access radius-profile radius-profile accounting radius-server-profile-name radius
commit
```

The RADIUS profile configuration is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config access radius-
profile radius-profile
{
  "rtbrick-config:radius-profile": [
    {
      "profile-name": "radius-profile",
      "nas-ip-address": "192.0.2.74",
      "nas-port-format": "DEFAULT",
      "nas-port-type": "Ethernet",
      "authentication": {
        "radius-server-profile-name": [
          "radius"
          ]
      },
      "accounting": {
        "radius-server-profile-name": [
          "radius"
          ]
      }
    }
  ]
}
```

e. Configure the RADIUS server `radius`.

```
set access radius-server radius
set access radius-server radius address 192.0.2.34
set access radius-server radius source-address 192.0.2.74
set access radius-server radius secret-encrypted-text
$2b2feb12f730107454b1be6a0f8242b0f
set access radius-server radius routing-instance default
set access radius-server radius rate 300
set access radius-server radius authentication enable true
set access radius-server radius authentication retry 3
set access radius-server radius authentication timeout 5
set access radius-server radius accounting enable true
set access radius-server radius accounting timeout 30
set access radius-server radius coa enable true
commit
```



The attribute `secret-plain-text` is converted to `secret-encrypted-text` in the show

> command output and the value is hashed.

The RADIUS server configuration is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config access radius-
server radius
{
  "rtbrick-config:radius-server": [
    {
      "server-name": "radius",
      "address": "192.0.2.34",
      "source-address": "192.0.2.74",
      "secret-encrypted-text": "$2b2feb12f730107454b1be6a0f8242b0f",
      "routing-instance": "default",
      "rate": 300,
      "authentication": {
        "enable": "true",
        "retry": 3,
        "timeout": 5
      },
      "accounting": {
        "enable": "true",
        "timeout": 30
      },
      "coa": {
        "enable": "true"
      }
    }
  ]
}
```

6. Configure the IPv4 and IPv6 access pools.

```
set access pool poolv4
set access pool poolv4 ipv4-address low 203.0.113.1
set access pool poolv4 ipv4-address high 203.0.113.64
set access pool poolv6
set access pool poolv6 ipv6-prefix low 2001:db8:0:1::/64
set access pool poolv6 ipv6-prefix high 2001:db8:0:2::/64
set access pool poolv6pd
set access pool poolv6pd ipv6-prefix low 2001:db8:0:3::/56
set access pool poolv6pd ipv6-prefix high 2001:db8:0:4::/56
commit
```

The access pool configuration is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config access pool
{
```

```
  "rtbrick-config:pool": [
    {
      "pool-name": "poolv4",
      "ipv4-address": {
        "low": "203.0.113.1",
        "high": "203.0.113.64"
      }
    },
    {
      "pool-name": "poolv6",
      "ipv6-prefix": {
        "low": "2001:db8:0:1::/64",
        "high": "2001:db8:0:2::/64"
      }
    },
    {
      "pool-name": "poolv6pd",
      "ipv6-prefix": {
        "low": "2001:db8:0:3::/56",
        "high": "2001:db8:0:4::/56"
      }
    }
  ]
}
```

3.4.2. IPoE Subscriber Management Configuration

IP-over-Ethernet (IPoE) is an access technology that uses DHCP for IPv4 and DHCPv6 for IPv6 where both protocols are handled in the IPoE daemon (`ipoed`). IPoE subscribers are identified by IFP, VLANs and client MAC addresses.

The dynamic creation of IPoE subscribers is triggered by DHCPv4 discover or DHCPv6 solicit request from the subscriber. Response is postponed until the subscriber is successfully authenticated using the known authentication methods such as local or RADIUS, however authentication is not mandatory. After the authentication phase, IPv4/IPv6/IPv6-PD address is allocated to the subscriber either from the local pool or from RADIUS.

To enable IPoE Subscriber Management, the following configurations are mandatory:

1. Access Interface Configuration
2. Access Profile Configuration
3. AAA (Authentication, Authorization and Accounting) Profile Configuration. Based on the authentication requirement, configure any one of the following:
    a. Local Authentication
        i. Pool Configuration
        ii. User Profile Configuration
    b. RADIUS Authentication

i. RADIUS Profile Configuration

ii. RADIUS Server Configuration

This solution section discusses RADIUS authentication.

NOTES:

- Access interfaces can be configured without VLAN tags (untagged) and with one (single tagged) or two (double tagged) VLAN tags.
- There can be more than one interface configured for subscriber management and each interface can reference the same profile.

Configuring IPoE Subscriber Management

For detailed information about the subscriber configuration options, see the Subscriber Management Configuration Guide.

1. Configure the access profile `ipoe`.

```
set access access-profile ipoe
set access access-profile ipoe protocol dhcp enable true
set access access-profile ipoe protocol dhcp lease-time 60
set access access-profile ipoe protocol dhcpv6 enable true
set access access-profile ipoe protocol dhcpv6 lifetime 60
set access access-profile ipoe address-family ipv4 enable true
set access access-profile ipoe address-family ipv4 pool-name poolv4
set access access-profile ipoe address-family ipv4 instance default
set access access-profile ipoe address-family ipv4 dad-enable true
set access access-profile ipoe address-family ipv6 enable true
set access access-profile ipoe address-family ipv6 pool-name poolv6
set access access-profile ipoe address-family ipv6 prefix-delegation-pool-name
poolv6pd
set access access-profile ipoe address-family ipv6 instance default
set access access-profile ipoe address-family ipv6 dad-enable true
commit
```

The access profile configuration is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show config access access-
profile
{
  "rtbrick-config:access-profile": [
    {
      "profile-name": "ipoe",
      "protocol": {
        "dhcp": {
          "enable": "true",
          "lease-time": 60
        },
```

```
        "dhcpv6": {
          "enable": "true",
          "lifetime": 60
        }
      },
      "address-family": {
        "ipv4": {
          "enable": "true",
          "pool-name": "poolv4",
          "instance": "default",
          "dad-enable": "true"
        },
        "ipv6": {
          "enable": "true",
          "pool-name": "poolv6",
          "prefix-delegation-pool-name": "poolv6pd",
          "instance": "default",
          "dad-enable": "true"
        }
      }
    }
  ]
}
```

2. Configure the Authentication and Accounting (AAA) profile for `aaa-profile`.

```
set access aaa-profile aaa-profile
set access aaa-profile aaa-profile session-timeout 0
set access aaa-profile aaa-profile idle-timeout 0
set access aaa-profile aaa-profile aaa-radius-profile radius-profile
set access aaa-profile aaa-profile authentication order RADIUS
set access aaa-profile aaa-profile accounting order RADIUS
set access aaa-profile aaa-profile accounting interim-interval 86400
set access aaa-profile aaa-profile accounting ingress accounting-source POLICER
set access aaa-profile aaa-profile accounting egress accounting-source CLASS
commit
```

The access AAA configuration is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show config access aaa-profile
{
  "rtbrick-config:aaa-profile": [
    {
      "profile-name": "aaa-profile",
      "session-timeout": 0,
      "idle-timeout": 0,
      "aaa-radius-profile": "radius-profile",
      "authentication": {
        "order": "RADIUS"
```

```
      },
      "accounting": {
        "order": "RADIUS",
        "interim-interval": 86400,
        "ingress": {
          "accounting-source": "POLICER"        },
        "egress": {
          "accounting-source": "CLASS"
        }
      }
    }
  ]
}
supervisor@rtbrick>multiservice-edge.rtbrick.net: op>
```

3. Configure the access interface. Double-tagged interface is configured in this case as the access interface (`ifp-0/1/5`). The interface configuration assigns the access type, access profile, AAA profile, and further optional attributes like service-profile to the specified access interface.

```
set access interface double-tagged ifp-0/1/5 1001 1100 1001 1100
set access interface double-tagged ifp-0/1/5 1001 1100 1001 1100 access-type IPoE
set access interface double-tagged ifp-0/1/5 1001 1100 1001 1100 access-profile-name
ipoe
set access interface double-tagged ifp-0/1/5 1001 1100 1001 1100 service-profile-name
subs_service
set access interface double-tagged ifp-0/1/5 1001 1100 1001 1100 aaa-profile-name aaa-
profile
set access interface double-tagged ifp-0/1/5 1001 1100 1001 1100 gateway-ifl lo-
0/0/1/0
commit
```

The double-tagged access interface configuration is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show config access interface
{
  "rtbrick-config:interface": {
    "double-tagged": [
      {
        "interface-name": "ifp-0/1/5",
        "outer-vlan-min": 1001,
        "outer-vlan-max": 1100,
        "inner-vlan-min": 1001,
        "inner-vlan-max": 1100,
        "access-type": "IPoE",
        "access-profile-name": "ipoe",
        "service-profile-name": "subs_service",
        "aaa-profile-name": "aaa-profile",
        "gateway-ifl": "lo-0/0/1/0"
```

```
      }
    ]
  }
}
```

4. In this solution, we configure AAA authentication and accounting with RADIUS. To use RADIUS authentication and accounting both the RADIUS profile and RADIUS server configurations (see below) must be configured.

a. Configure RADIUS profile `radius-profile`.

```
set access radius-profile radius-profile
set access radius-profile radius-profile nas-ip-address 192.0.2.74
set access radius-profile radius-profile nas-port-format DEFAULT
set access radius-profile radius-profile nas-port-type Ethernet
set access radius-profile radius-profile authentication radius-server-profile-name
radius
set access radius-profile radius-profile accounting radius-server-profile-name radius
commit
```

The RADIUS profile configuration is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show config access radius-
profile radius-profile
{
  "rtbrick-config:radius-profile": [
    {
      "profile-name": "radius-profile",
      "nas-ip-address": "192.0.2.74",
      "nas-port-format": "DEFAULT",
      "nas-port-type": "Ethernet",
      "authentication": {
        "radius-server-profile-name": [
          "radius"
          ]
      },
      "accounting": {
        "radius-server-profile-name": [
          "radius"
          ]
      }
    }
  ]
}
```

b. Configure the RADIUS server `radius`.

```
set access radius-server radius
set access radius-server radius address 192.0.2.34
set access radius-server radius source-address 192.0.2.74
set access radius-server radius secret-plain-text testing123
set access radius-server radius routing-instance default
set access radius-server radius rate 300
set access radius-server radius authentication enable true
set access radius-server radius authentication retry 3
set access radius-server radius authentication timeout 5
set access radius-server radius accounting enable true
set access radius-server radius accounting timeout 30
set access radius-server radius coa enable true
commit
```



The attribute` secret-plain-text` will be converted to `secret-encrypted-text` in the show command output and value will be hashed.

The RADIUS server configuration is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config access radius-
server radius
{
  "rtbrick-config:radius-server": [
    {
      "server-name": "radius",
      "address": "192.0.2.34",
      "source-address": "192.0.2.74",
      "secret-encrypted-text": "$2b2feb12f730107454b1be6a0f8242b0f",
      "routing-instance": "default",
      "rate": 300,
      "authentication": {
        "enable": "true",
        "retry": 3,
        "timeout": 5
      },
      "accounting": {
        "enable": "true",
        "timeout": 30
      },
      "coa": {
        "enable": "true"
      }
    }
  ]
}
```

3. Configure the IPv4 and IPv6 access pools.

```
set access pool poolv4
set access pool poolv4 ipv4-address low 203.0.113.1
set access pool poolv4 ipv4-address high 203.0.113.64
set access pool poolv6
set access pool poolv6 ipv6-prefix low 2001:db8:0:1::/64
set access pool poolv6 ipv6-prefix high 2001:db8:0:40::/64
set access pool poolv6pd
set access pool poolv6pd ipv6-prefix low 2001:db8:0:100::/56
set access pool poolv6pd ipv6-prefix high 2001:db8:0:4000::/56
commit
```

The access pool configuration is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config access pool
{
  "rtbrick-config:pool": [
    {
      "pool-name": "poolv4",
      "ipv4-address": {
        "low": "203.0.113.1",
        "high": "203.0.113.64"
      }
    },
    {
      "pool-name": "poolv6",
      "ipv6-prefix": {
        "low": "2001:db8:0:1::/64",
        "high": "2001:db8:0:40::/64"
      }
    },
    {
      "pool-name": "poolv6pd",
      "ipv6-prefix": {
        "low": "2001:db8:0:100::/56",
        "high": "2001:db8:0:4000::/56"
      }
    }
  ]
}
```

3.4.3. Quality of Service (QoS) Configuration

Following are the steps involved in configuring and verifying IPoE and PPPoE QoS:

1. Enabling System Platform Profile for Single Queue Subscribers
2. Configuring service profile to enable QoS on subscriber
3. Configuring downstream QoS

4. Configuring upstream QoS
5. Configuring PPPoE and IPoE subscriber accounting for upstream and downstream traffic
6. Configuring PPPoE and IPoE subscriber QoS on BNG Blaster
7. Validating PPPoE and IPoE QoS on BNG Blaster

The figure below shows how QoS is configured for ingress and egress traffic.

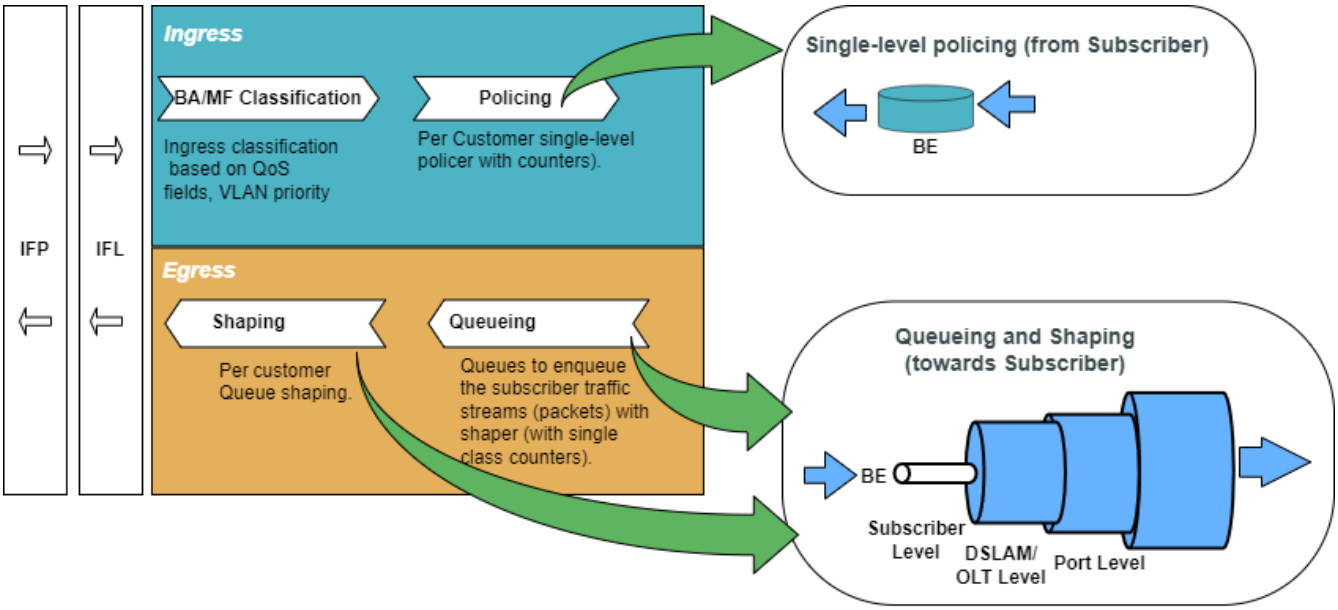


Figure 13. QoS Cnfiguration for Ingress and Egress Traffic

For detailed information about the QoS configuration options, see HQoS Configuration Guide.

Enabling System Platform Profile for Single Q Subscribers

System reboot is required after enabling the system platform profile.

1. Enable system platform profile for single queue subscribers

```
set system platform profile nat_1q
commit
```

The configuration of the system platform profile named `nat_1q` is shown below.

```
supervisor@rtbrick>ec24.q2c.u9.r2.nbg.rtbrick.net: cfg> show config system platform
profile
{
  "rtbrick-config:profile": "nat_1q"
}
```

Configure Service Profile

Service profile configuration in subscriber management allows to assign QoS configurations to a

subscriber.

1. Configure the service profile to enable QoS. The service profile defined to enable Quality of Service with the profile name is `residential`.

```
set access service-profile subs_service qos profile residential
commit
```

The configuration of the service profile named `residential` is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config access service-
profile subs_service
{
  "rtbrick-config:service-profile": [
    {
      "profile-name": "subs_service",
      "qos": {
        "profile": "residential"
      }
    }
  ]
}
```

2. Enable QoS on the PPPoE and IPoE subscriber access interface (`ifp-0/1/3` and `ifp-0/1/5` respectively) to enable QoS for the subscriber.

```
set access interface double-tagged ifp-0/1/3 1001 1100 1001 1100 serviceprofile-name
subs_service
set access interface double-tagged ifp-0/1/5 1001 1100 1001 1100 serviceprofile-name
subs_service
commit
```

Below is the double-tagged access interface on which the service profile `subs_service` is configured for access type PPPoE:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config access interface
double-tagged ifp-0/1/3
{
  "rtbrick-config:double-tagged": [
    {
      "interface-name": "ifp-0/1/3",
      "outer-vlan-min": 1001,
      "outer-vlan-max": 1100,
      "inner-vlan-min": 1001,
      "inner-vlan-max": 1100,
      "access-type": "PPPoE",
      "service-profile-name": "subs_service",
```

```
      "aaa-profile-name": "aaa-profile",
      "gateway-ifl": "lo-0/0/1/0"
    }
  ]
}
```

Below is the double-tagged access interface on which the service profile `subs_service` is configured for access type IPoE:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config access interface
double-tagged ifp-0/1/5
{
  "rtbrick-config:double-tagged": [
    {
      "interface-name": "ifp-0/1/5",
      "outer-vlan-min": 1001,
      "outer-vlan-max": 1100,
      "inner-vlan-min": 1001,
      "inner-vlan-max": 1100,
      "access-type": "IPoE",
      "service-profile-name": "subs_service",
      "aaa-profile-name": "aaa-profile",
      "gateway-ifl": "lo-0/0/1/0"
    }
  ]
}
```

3. Configure QoS profile to enable on subscriber.

```
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-
1queue
set forwarding-options class-of-service profile residential class-policer-map-name
policer-map-single
set forwarding-options class-of-service profile residential policer-name policer-
single
set forwarding-options class-of-service profile residential scheduler-map-name subs-
1queue-single-play
commit
```

The QoS Profile with all the primitives needed to enable traffic profiles on subscribers are as follows:

```
supervisor@rtbrick>ufi16.q2c.u13.r5.nbg.rtbrick.net: cfg> show config forwarding-
options class-of-service profile residential
```

```
{
  "rtbrick-config:profile": [
    {
      "profile-name": "residential",
      "classifier-name": "subs-pbit-class",
      "class-queue-map-name": "subs-1queue",
      "class-policer-map-name": "policer-map-single",
      "policer-name": "policer-single",
      "scheduler-map-name": "subs-1queue-single-play"
    }
  ]
}
```

Configure Downstream QoS

Downstream Quality of Service (QoS) is used to prioritize network traffic from the Internet to subscribers.

1. Enable global classification for downstream traffic.

```
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 source-ipv4-prefix 192.0.2.2/32
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1001 action forward-class class-0
commit
```

Below is the multi-field-classifier (MFC) based classifier for global enabling of downstream traffic classification.

```
supervisor@rtbrick>ufi16.q2c.u13.r5.nbg.rtbrick.net: cfg> show config forwarding-
options class-of-service multifield-classifier
{
  "rtbrick-config:multifield-classifier": {
    "acl": {
      "l3v4": {
        "rule": [
          {
            "rule-name": "global_mfc",
            "ordinal": [
              {
                "ordinal-value": 1001,
                "match": {
                  "source-ipv4-prefix": "192.0.2.2/32"
                },
                "action": {
                  "forward-class": "class-0"
```

```
              }
            }
          ]
        }
      ]
    }
  }
}
```

2. Configure the MFC-based classifier with qualifiers and actions.

```
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 192.0.2.2/32
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 192.0.2.2/32
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1002 action forward-class class-1
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1003
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1003 match ipv4-tos 192
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1003 match source-ipv4-prefix 192.0.2.2/32
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1003 action forward-class class-2
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1004
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1004 match ipv4-tos 224
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1004 match source-ipv4-prefix 192.0.2.2/32
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1004 action forward-class class-3
commit
```

The configuration of the QoS MFC-based Classifier for the classification of downstream traffic from the core towards Subscriber is shown below.

```
supervisor@rtbrick>multiservice-edge.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": "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"
          }
        }
      ]
    }
  ]
```

```
}
```

3. Configure the queues needed for enqueuing and dequeuing traffic streams.

```
set forwarding-options class-of-service queue BE_SUBS1
set forwarding-options class-of-service queue BE_SUBS1 queue-size 375000
set forwarding-options class-of-service queue BE_SUBS1 shaper-name shaper_BE
set forwarding-options class-of-service queue BE_SUBS1 header-compensation bytes 22
set forwarding-options class-of-service queue BE_SUBS1 header-compensation decrement
true
commit
```

The queue configuration is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config forwarding-options
class-of-service queue
{
  "rtbrick-config:queue": [
    {
      "queue-name": "BE_SUBS1",
      "queue-size": 375000,
      "shaper-name": "shaper_BE",
      "header-compensation": {
        "bytes": 22,
        "decrement": "true"
      }
    }
  ]
}
```

4. Configure downstream traffic shaping for the queue.



Queue Shaping is only on the BE_SUBS1 queue.

```
set forwarding-options class-of-service shaper shaper_BE
set forwarding-options class-of-service shaper shaper_BE shaping-rate-high 2000
set forwarding-options class-of-service shaper shaper_BE shaping-rate-low 0
commit
```

The shaping Configuration is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config forwarding-options
class-of-service shaper
{
  "rtbrick-config:shaper": [
    {
```

```
      "shaper-name": "shaper_BE",
      "shaping-rate-high": 2000,
      "shaping-rate-low": 0
    }
  ]
}
```

5. Enqueue classified traffic to a queue using class-to-queue mapping.

```
set forwarding-options class-of-service class-queue-map subs-1queue
set forwarding-options class-of-service class-queue-map subs-1queue class class-0
set forwarding-options class-of-service class-queue-map subs-1queue class class-0
queue-name BE_SUBS1
commit
```

Class-to-Queue mapping configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config forwarding-options
class-of-service class-queue-map subs-1queue
{
  "rtbrick-config:class-queue-map": [
    {
      "class-queue-map-name": "subs-1queue",
      "class": [
        {
          "class-type": "class-0",
          "queue-name": "BE_SUBS1"
        }
      ]
    }
  ]
}
```

Configure Upstream QoS

1. Configure the BA Classifier for the classification of multiple traffic streams targeted at PPPoE and IPoE subscribers:

```
set forwarding-options class-of-service classifier subs-pbit-class
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 1
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 1 class class-0
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 2
```

```
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 2 class class-0
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 3
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 3 class class-0
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 4
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 4 class class-0
commit
```

The configuration of the QoS BA-based Classifier for the classification of upstream traffic toward the PPPoE Subscriber is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config forwarding-options
class-of-service classifier subs-pbit-class
{
  "rtbrick-config:classifier": [
    {
      "classifier-name": "subs-pbit-class",
      "match-type": [
        {
          "match-type": "ieee-802.1",
          "codepoint": [
            {
              "codepoint": 1,
              "class": "class-0"
            },
            {
              "codepoint": 2,
              "class": "class-0"
            },
            {
              "codepoint": 3,
              "class": "class-0"
            },
            {
              "codepoint": 4,
              "class": "class-0"
            }
          ]
        }
      ]
    }
  ]
}
```

2. Configure single-level policer to police Level1 traffic.

```
set forwarding-options class-of-service policer policer-single
set forwarding-options class-of-service policer policer-single level1-rates cir 2000
set forwarding-options class-of-service policer policer-single level1-rates cbs 1000
set forwarding-options class-of-service policer policer-single level1-rates pir 2500
set forwarding-options class-of-service policer policer-single level1-rates pbs 1000
set forwarding-options class-of-service policer policer-single mode single-level
commit
```

The single-level policer configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net:: cfg> show config forwarding-options
class-of-service policer
{
  "rtbrick-config:policer": [
    {
      "policer-name": "policer-single",
      "level1-rates": {
        "cir": 2000,
        "cbs": 1000,
        "pir": 2500,
        "pbs": 1000
      },
      "mode": "single-level"
    }
  ]
}
s
```

3. Map the classified traffic streams to a single policer level using class-to-policer mapping:

```
set forwarding-options class-of-service class-policer-map policer-map-single
set forwarding-options class-of-service class-policer-map policer-map-single class
class-0
set forwarding-options class-of-service class-policer-map policer-map-single class
class-0 policer-level level-1
commit
```

The class-policer-map configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config forwarding-options
class-of-service class-policer-map
{
  "rtbrick-config:class-policer-map": [
    {
      "class-policer-map-name": "policer-map-single",
      "class": [
        {
```

```
          "class": "class-0",
          "policer-level": "level-1"
        }
      ]
    }
  ]
}
```

3.4.4. Configure FreeRADIUS Server

Installation of FreeRADIUS

FreeRADIUS server can be installed on any Linux OS distribution. For information about installing FreeRADIUS, see Installing the FreeRADIUS Server.

Remove the Unsupported Files

It is necessary to remove `echo`, `ntlm_auth`, `eap`, `echo`, and `mschap` files once FreeRadius has been installed, since they are not required by this reference design. To remove these files, enter the following commands:

```
rm -rf /etc/freeradius/3.0/mods-enabled/echo
rm -rf /etc/freeradius/3.0/mods-enabled/ntlm_auth
rm -rf /etc/freeradius/3.0/mods-enabled/eap
rm -rf /etc/freeradius/3.0/mods-enabled/echo
rm -rf /etc/freeradius/3.0/mods-enabled/mschap
```

FreeRADIUS User and Group Settings

Using the following commands, one can set the user or group as root.

```
sed -i 's/User=freerad/User=root/g' /lib/systemd/system/freeradius.service
sed -i 's/Group=freerad/Group=root/g' /lib/systemd/system/freeradius.service
```

Run the command for reloading 'systemd' after user or group settings:

```
systemctl daemon-reload
```

Configure the FreeRADIUS Files

It is necessary to configure the FreeRADIUS files once FreeRadius has been installed. The configuration files can be found under `/etc/freeradius/3.0/`.

authorize

Using the following command, one can view the `authorize` file in its default location.

```
~:/etc/freeradius/3.0 # cat mods-config/files/authorize
```

Replace the content of the `authorize` file with the following:

```
$INCLUDE /etc/freeradius/3.0/pppoe_users_file
```



This file can also be downloaded from the appendix (Appendix C: RADIUS Server Configuration).

pppoe_ipoe_file_16k

The IPoE and PPPoE users file (`pppoe_ipoe_file_16k`) mentioned in the above section includes subscriber profile parameters as shown below.

```
# users.py --temlate templates/user.j2 --user user{i}@rtbrick.com --count 25000 --bng
BNG --ipv4 10.100.128.1 --ipv6 10:1:1::0 --ipv6pd 50:1:1::0 --dns-primary 10.0.0.3
--dns-secondary 10.0.0.4 --dns-ipv6-primary fc66:0::3 --dns-ipv6-secondary fc66:0::4
DEFAULT User-Name =~ '^[0-9a-f\:]+@ipoe$', Cleartext-Password := 'ipoe'
    Service-Type = Framed-User,
    Class = "IPOE",
    Reply-Message = "FOOBAR Internet",
    RtBrick-DNS-Primary = 10.0.0.3,
    RtBrick-DNS-Secondary = 10.0.0.4,
    RtBrick-DNS-Primary-IPv6  = fc66:0::3,
    RtBrick-DNS-Secondary-IPv6 = fc66:0::4,
    Acct-Interim-Interval = 60,
    RtBrick-QoS-Profile = "subs-triple-play"
"user1@rtbrick.com" Cleartext-Password := "test"
    Service-Type = Framed-User,
    Framed-Protocol = PPP,
    Class = "user1@rtbrick.com",
    Framed-IP-Address = 10.100.128.1,
    Framed-IP-Netmask = 255.255.255.255,
    RtBrick-DNS-Primary = 10.0.0.3,
    RtBrick-DNS-Secondary  = 10.0.0.4,
    Framed-IPv6-Prefix = 70:20:2::/64,
    Delegated-IPv6-Prefix = 70:2:2::/56,
    RtBrick-DNS-Primary-IPv6  = fc66:0::3,
    RtBrick-DNS-Secondary-IPv6 = fc66:0::4,
    Session-Timeout = 0,
    Idle-Timeout = 0,
    Acct-Interim-Interval = 60,
```

```
    RtBrick-QoS-Profile = "subs-triple-play"
"user2@rtbrick.com" Cleartext-Password := "test"
    Service-Type = Framed-User,
    Framed-Protocol = PPP,
    Class = "user2@rtbrick.com",
    Framed-IP-Address = 10.100.128.2,
    Framed-IP-Netmask = 255.255.255.255,
    RtBrick-DNS-Primary = 10.0.0.3,
    RtBrick-DNS-Secondary  = 10.0.0.4,
    Framed-IPv6-Prefix = fc75:1:1:1::/64,
    Delegated-IPv6-Prefix = fc76:1:1:100::/56,
    RtBrick-DNS-Primary-IPv6  = fc66:0::3,
    RtBrick-DNS-Secondary-IPv6 = fc66:0::4,
    Session-Timeout = 0,
    Idle-Timeout = 0,
    Acct-Interim-Interval = 60,
    RtBrick-QoS-Profile = "subs-triple-play"
"user3@rtbrick.com" Cleartext-Password := "test"
    Service-Type = Framed-User,
    Framed-Protocol = PPP,
    Class = "user3@rtbrick.com",
    Framed-IP-Address = 10.100.128.3,
    Framed-IP-Netmask = 255.255.255.255,
    RtBrick-DNS-Primary = 10.0.0.3,
    RtBrick-DNS-Secondary  = 10.0.0.4,
    Framed-IPv6-Prefix = fc75:1:1:2::/64,
    Delegated-IPv6-Prefix = fc76:1:1:200::/56,
    RtBrick-DNS-Primary-IPv6  = fc66:0::3,
    RtBrick-DNS-Secondary-IPv6 = fc66:0::4,
    Session-Timeout = 0,
    Idle-Timeout = 0,
    Acct-Interim-Interval = 60,
    RtBrick-QoS-Profile = "subs-triple-play"
"user4@rtbrick.com" Cleartext-Password := "test"
    Service-Type = Framed-User,
    Framed-Protocol = PPP,
    Class = "user4@rtbrick.com",
    Framed-IP-Address = 10.100.128.4,
    Framed-IP-Netmask = 255.255.255.255,
    RtBrick-DNS-Primary = 10.0.0.3,
    RtBrick-DNS-Secondary  = 10.0.0.4,
    Framed-IPv6-Prefix = fc75:1:1:3::/64,
    Delegated-IPv6-Prefix = fc76:1:1:300::/56,
    RtBrick-DNS-Primary-IPv6  = fc66:0::3,
    RtBrick-DNS-Secondary-IPv6 = fc66:0::4,
    Session-Timeout = 0,
    Idle-Timeout = 0,
    Acct-Interim-Interval = 60,
    RtBrick-QoS-Profile = "subs-triple-play"
    <...>
```

The `pppoe_ipoe_file_16k` can be created with the above content in the `/etc/freeradius/3.0/` path. Alternatively, this file can be downloaded from the appendix section of this guide and placed at `/etc/freeradius/3.0/`.

clients.conf

Clients.conf file shall be configured with the expected RADIUS client IP address and secret.

```
~:/etc/freeradius/3.0 # cat clients.conf
client rtbrick {
        ipaddr          = 192.0.2.74
        secret          = testing123
        shortname       = rtbrick
        nas_type        = other
        require_message_authenticator = no
}
```

The `clients.conf` file (/etc/freeradius/3.0/clients.conf) used for this reference design can be downloaded from the appendix section of this guide.

radiusd.conf

The `radiusd.conf` file should be configured with the expected RADIUS authentication and accounting parameters.

```
prefix = /usr
exec_prefix = /usr
sysconfdir = /etc
localstatedir = /var
sbindir = ${exec_prefix}/sbin
logdir = /var/log/freeradius
raddbdir = /etc/freeradius
radacctdir = ${logdir}/radacct
name = freeradius
confdir = ${raddbdir}
modconfdir = ${confdir}/mods-config
run_dir = ${localstatedir}/run/${name}
db_dir = ${raddbdir}
libdir = /usr/lib/freeradius
pidfile = ${run_dir}/${name}.pid
correct_escapes = true
max_request_time = 5
cleanup_delay = 0
max_requests = 16384
hostname_lookups = no
log {
    destination = files
    file = ${logdir}/radius.log
```

```
    stripped_names = no
    auth = yes
}
checkrad = ${sbindir}/checkrad
security {
    # user = radius
    # group = radius
    allow_core_dumps = no
    max_attributes = 200
    reject_delay = 0
    status_server = no
@openssl_version_check_config@
}
proxy_requests  = no
$INCLUDE clients.conf
thread pool {
    start_servers = 32
    max_servers = 128
    min_spare_servers = 8
    max_spare_servers = 16
    max_queue_size = 16384
    max_requests_per_server = 0
    auto_limit_acct = no
}
modules {
    $INCLUDE mods-enabled/
}
instantiate {
    files
    linelog
}
server default {
    listen {
        type = auth
        ipaddr = *
        port = 1812
    }
    listen {
        type = acct
        ipaddr = *
        port = 1813
    }
    authorize {
        update request {
            FreeRADIUS-Client-Shortname = "%{&request:Client-Shortname}"
        }
        if (&request:Client-Shortname == "rtbrick-server") {
            rtbrick-server-log
        }
        files
        pap
```

```
    }
    authenticate {
        pap
    }
    post-auth {
        if (&request:Client-Shortname == "rtbrick-server") {
            rtbrick-server-log
        }
        Post-Auth-Type REJECT {
            update reply {
                Reply-Message := "Login Failed. Please check your username and
password."
            }
            attr_filter.access_reject
        }
    }
    preacct {
        ok
    }
    accounting {
        if (&request:Client-Shortname == "rtbrick-server") {
            rtbrick-server-log
        }
    ok
    }
    session {
    }
}
```

Radiusd.conf should be configured to use UDP ports 1812 and 1813 for authentication and accounting, respectively. Additionally, `rtbrick-server-log` should be added to the parameters for authorise, authenticate, and accounting.

The `radiusd.conf` file (/etc/freeradius/3.0/radiusd.conf) used for this reference design can be downloaded from the appendix section (Appendix C) of this guide.

detail

The `detail` file shall be configured for the RADIUS accounting logs.

```
/etc/freeradius/3.0 # cat mods-enabled/detail
    permissions = 0666detail rtbrick-server-log {
    filename = ${radacctdir}/rtbrick-server-detail.log

    header = "%t;%{NAS-IP-Address};%I;%{Packet-Src-Port}"
    log_packet_header = no
}
```

Ensure that `rtbrick-server-log` is specified in the `detail` file.

The `detail` file (/etc/freeradius/3.0/mods-enabled/detail) used for this reference design can be downloaded from the appendix section (Appendix C) of this guide.

dictionary.rtbrick

Add the RtBrick RADIUS dictionary (dictionary.rtbrick) to `/usr/share/freeradius/dictionary.rtbrick` and include it in ` /usr/share/freeradius/dictionary `.

The `dictionary.rtbrick` contains the RBFS attributes in FreeRADIUS format.

- Click here to download the `radius_config.zip`, which contains the `dictionary.rtbrick` file.

Stopping and Starting the FreeRADIUS Server for any Changes

For any changes, stop and restart the FreeRADIUS server.

To stop the server, enter the following command:

```
sudo service freeradius stop
```

To start the server, enter the following command:

```
sudo service freeradius start
```

The FreeRadius server is now ready to provide AAA (Authentication, Accounting & Authorization) services to logging in subscribers.

Displaying the RADIUS Service Status

Run the following command to determine whether RADIUS service is active:

```
sudo service freeradius status
```

3.5. CGNAT Configuration

3.5.1. Configure CGNAT

To deploy CGNAT functionalities in an RBFS device, you must complete the following configurations:

- NAT Profile
- NAT Pool
- NAT Service Profile Configuration

- Enable NAT Service Profile on the Access Interface

The following sections provide the commands to configure RBFS CGNAT. For more information about CGNAT configuration, see *CGNAT Configuration Guide*.

NAT Profile Configuration

Run the following commands to configure NAT profile. A NAT profile defines how the NAT device has to perform the IPv4 address translation. You can create NAT profile for an RBFS instance using the 'instance' option. Also, it allows you to define the TCP or UDP traffic type for the profile.

```
set forwarding-options address-translation profile profile1
set forwarding-options address-translation profile profile1 instance default
set forwarding-options address-translation profile profile1 pool pool1
set forwarding-options address-translation profile profile1 max-rules 256
set forwarding-options address-translation profile profile1 ip-protocol tcp ageing-
timeout 120
set forwarding-options address-translation profile profile1 ip-protocol udp ageing-
timeout 300
set forwarding-options address-translation profile profile1 ip-protocol other ageing-
timeout 300
```

NAT profile configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config forwarding-options
address-translation profile
{
  "rtbrick-config:profile": [
    {
      "profile": "profile1",
      "instance": "default",
      "pool": "pool1",
      "max-rules": "256",
      "ip-protocol": {
        "tcp": {
          "ageing-timeout": 120
        },
        "udp": {
          "ageing-timeout": 300
        },
        "other": {
          "ageing-timeout": 300
        }
      }
    }
  ]
}
```

NAT Pool Configuration

Run the following commands to create IPv4 address pools. A pool includes a range of public IPv4 addresses. You can define the group of public IPs belonging to that pool by specifying the lowest and highest IP addresses for the pool. The default port block size is 32, which cannot be changed.

```
set forwarding-options address-translation pool pool1
set forwarding-options address-translation pool pool1 ipv4-address low 100.100.100.100
set forwarding-options address-translation pool pool1 ipv4-address high
100.100.100.149
```

NAT Pool configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config forwarding-options
address-translation pool
{
  "rtbrick-config:pool": [
    {
      "pool-name": "pool1",
      "ipv4-address": {
        "low": "100.100.100.100",
        "high": "100.100.100.149"
      }
    }
  ]
}
```

NAT Service Profile Configuration

One 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.

Enable NAT Service Profile on the Access Interface

It is required to enable the NAT service profile created on the access interface. Run the following commands to enable the NAT service profile on the interface. A service profile named `subs_service` and a profile named `nat_profile1` are configured.

The following configuration attaches the access interface `ifp-0/1/40` with the service profile `subs_service` for the PPPoE and IPoE subscribers.

```
set access interface double-tagged ifp-0/1/40 1000 1007 84 4084
set access interface double-tagged ifp-0/1/40 1000 1007 84 4084 access-type PPPoE
set access interface double-tagged ifp-0/1/40 1000 1007 84 4084 access-profile-name
pppoe
set access interface double-tagged ifp-0/1/40 1000 1007 84 4084 service-profile-name
subs_service
set access interface double-tagged ifp-0/1/40 1000 1007 84 4084 aaa-profile-name ipoe-
aaa
set access interface double-tagged ifp-0/1/41 1000 1007 84 4084
set access interface double-tagged ifp-0/1/41 1000 1007 84 4084 access-type IPoE
set access interface double-tagged ifp-0/1/41 1000 1007 84 4084 access-profile-name
ipoe
set access interface double-tagged ifp-0/1/41 1000 1007 84 4084 service-profile-name
subs_service
set access interface double-tagged ifp-0/1/41 1000 1007 84 4084 aaa-profile-name ipoe-
aaa
set access interface double-tagged ifp-0/1/41 1000 1007 84 4084 gateway-ifl lo-
0/0/0/100
```

The configuration for attaching the service profile with the access interface is shown below:

```
supervisor@rtbrick>.rtbrick.net: op> show config access interface
{
  "rtbrick-config:interface": {
    "double-tagged": [
      {
        "interface-name": "ifp-0/1/40",
        "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": "subs_service",
        "aaa-profile-name": "ipoe-aaa"
      },
      {
        "interface-name": "ifp-0/1/41",
        "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": "subs_service",
        "aaa-profile-name": "ipoe-aaa",
```

```
        "gateway-ifl": "lo-0/0/0/100"
      }
    ]
  }
}
```

3.6. Appendixes

Appendix A: RBFS Multiservice Edge Configuration

The RBFS Multiservice Edge configuration file (`multiservice-edge-ospf-cgnat.json`) can be downloaded from here.

- Click here to download the RBFS Multiservice Edge configuration file.

Appendix B: TACACS+ Server Configuration

The TACACS+ server configuration file (`tac_plus.conf`) can be downloaded from here.

- Click here to download the TACACS+ server configuration file.

Appendix C: RADIUS Server Configuration

The RADIUS server configuration files (`radius_config.zip`) can be downloaded from here. The zip archive contains the set of configuration files needed to configure the RADIUS server.

- Click here to download the RADIUS server configuration files.

Appendix D: BNG Blaster Configuration

The BNG Blaster configuration file (`blaster_cgnat.json`) can be downloaded from here.

- Click here to download the BNG Blaster configuration files.

The JSON files (`ospfv2_3node.json` and `ospfv3_3node.json`) which are used to simulate R-1, R-2, and RR on BNG Blaster, can be downloaded from here.

- Click here to download the `ospfv2_3node.json` file.
- Click here to download the `ospfv3_3node.json` file.

4. RBFS Multiservice Edge with Redundancy for IPoE Subscribers

4.1. RBFS Multiservice Edge with Redundancy for IPoE Subscribers

RtBrick provides reference design architectures designed and validated with in-house testing tools and methods. This document guides you to validate RBFS Multiservice Edge - IPoE with Redundancy implementation.

4.1.1. About this Guide

This document provides information about the RBFS Multiservice Edge - IPoE deployment in Redundancy (high availability) mode. The guide contains information about general platform configuration, configuration of various access and routing protocols, subscriber management, Quality of Service (QoS) and so on. The document presents a single use case scenario and provides information specifically on how to validate this particular implementation and for more information on any specific application, refer to https://documents.rtbrick.com/.

Currently, the guide's scope is limited to the basic features and configurations for validation purposes. This guide does not provide information about the advanced RBFS features such as multicast.

4.1.2. About the RBFS Multiservice Edge

The RtBrick Multiservice Edge is delivered as a container, running on RtBrick Host provided by the hardware ODM manufacturers. Platforms that support Multiservice Edge include Edgecore AGR 400, CSR 320, and UfiSpace S9600. The RtBrick Multiservice Edge software runs on powerful bare-metal switches as an open BNG.

The BNG is designed to dynamically deliver the following services:

1. Discovering and managing subscriber sessions for IPoE subscribers
2. Providing authentication, authorization and accounting (AAA)

4.1.3. Deployment

A Multiservice Edge provides BNG functionality on a single bare-metal switch and eliminates the need to have a chassis based system. It provides a low footprint and optimal power consumption based on BRCM chipsets, a compelling value proposition that has complete BNG and routing feature support.

Multiservice Edge runs on small form-factor temperature hardened hardware that allows deployments in street site cabinets.

For more information, see the RBFS Installation Guide.

4.1.4. Using the RBFS CLI

Connect to the `multiservice-edge` node.

```
$ ssh <multiservice-edge-management-ip> -l supervisor
supervisor@<multiservice-edge-management-ip>'s password:
```

The password for `multiservice-edge-management-ip` should be entered here.

As a result, the CLI prompt will look like this:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net:~ $
```

Open the RBFS CLI.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net:~ $ cli
```

The CLI has three different modes:

- `operation` mode is a read-only mode to inspect and analyse the system state
- `config` mode allows modifying the RBFS configuration
- `debug` mode provides advanced tools for trouble-shooting

The `switch-mode` command allows switching between the different modes. The `show` commands allow inspecting the system state. The `set` and `delete` commands, which are only available in the configuration mode, allow modifying or deleting the current configuration. The `commit` command persists the changes. RBFS provides a commit history which allows reviewing changes (`show commit log`) and restoring a previous configuration (`rollback`). There are also commands to `ping` destinations, `capture` network traffic, `save` the configuration to or `load` the configuration from a file.

The CLI supports abbreviating commands, provides suggestions by hitting the `[tab]` key and displays a context help by entering a `?`.

For more information on how to use the RBFS CLI, see the RBFS CLI User Guide.

4.1.5. About RBFS Redundancy

RBFS Redundancy protects subscriber services from node and link outages. 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 RBFS Redundancy, if one node/protected link goes down, another node can automatically take over the services (with a minimal downtime) for a redundancy group.

RBFS Redundancy protects subscriber groups using an active-standby node cluster model. The active node for a redundancy group runs the workload, and the peer node, which acts as standby, mirrors the subscriber state data from the peer (active) and takes over the workload in the event of

a node or link failure. It ensures that traffic can keep flowing in the event of an outage. For more information about RBFS redundancy solution, see RBFS Redundancy Solution Guide.

4.1.6. Redundancy Pair BNG Devices and Inter-BNG Connectivity

Multiservice Edge platforms, paired for redundancy, are connected with a Redundancy (RD) TCP link. This RD TCP connection can be formed either directly or through the core network. It uses IS-IS (for unicast reachability) and LDP (for labeled unicast reachability) between the active and standby nodes. The link between the Multiservice Edge pairs establishes connectivity. RD 'keepalive' messages are exchanged between the two CBNG devices and happens data mirroring for subscriber state synchronization.

Both the devices which are paired for redundancy can contain multiple redundancy sessions. Multiple Multiservice Edge nodes can be active nodes for one or more redundancy sessions that serve the subscribers and they can, at the same time, be standby nodes for other subscriber groups.

4.1.7. RBFS Multiservice Edge with Redundancy for IPoE Subscribers Implementation Architecture

The following topology diagram for RBFS Multiservice Edge IPoE shows two bare metal switches installed with RBFS Multiservice Edge software. This topology aims to demonstrate complete IPoE subscriber emulation with redundancy and routing to connect to the network uplink.

The devices are deployed in active and standby mode with respect to the lag bundles. `cbng1` is configured as the active device for lag-1, and `cbng2` is the standby device for lag-1. If one node goes down, the other node becomes stand-alone and takes over the subscribers for a subscriber group. RBFS devices (CBNGs) are connected to core routers on one side and OLTs (simulated by BNGBlaster) on the other.

When the active device goes down or a link failure occurs between the active CBNG device and the OLT device, the standby CBNG device detects the same and takes over the IPoE subscriber sessions from the previously active device.

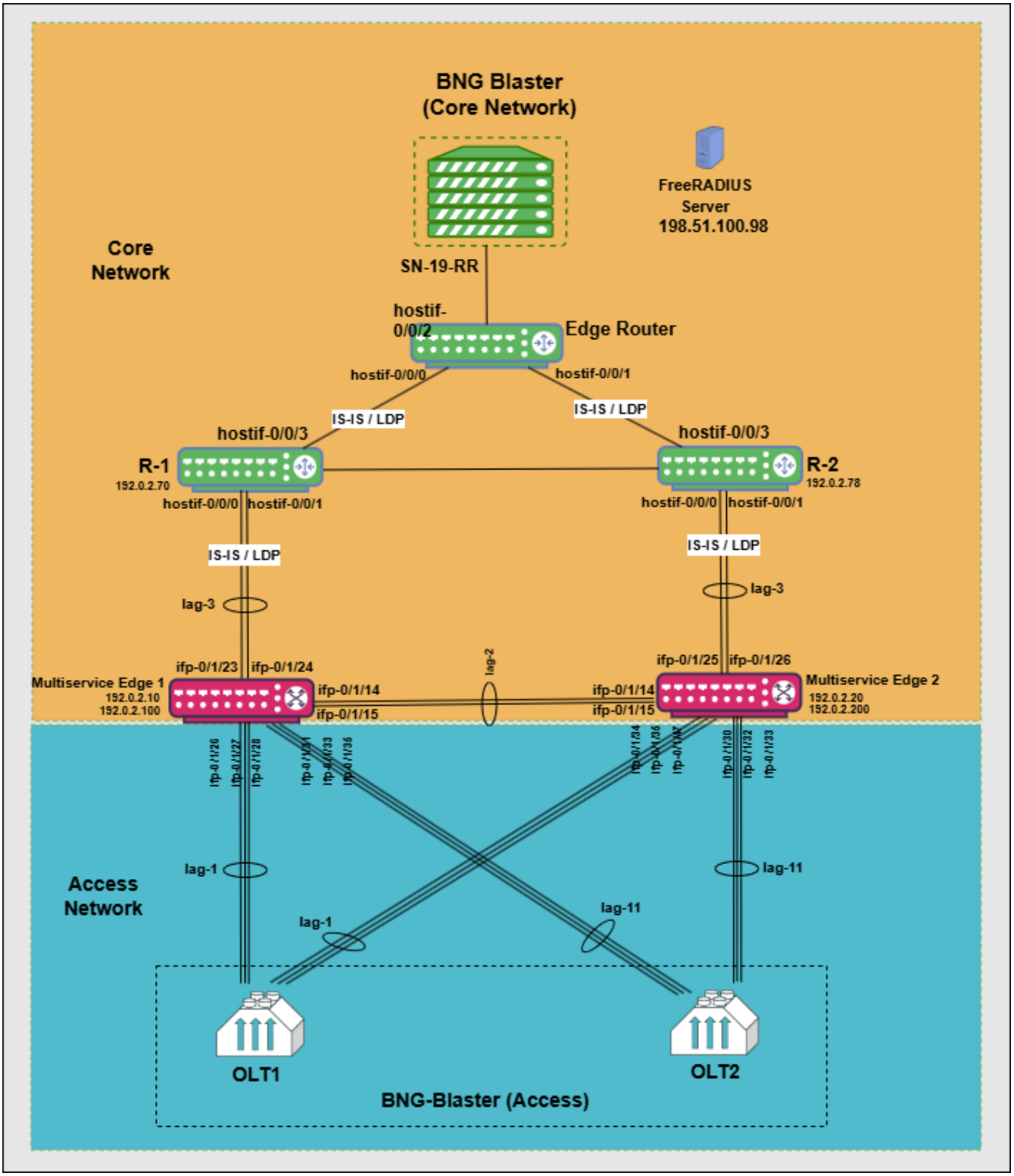


Figure 14. RBFS Multiservice Edge with Redundancy for IPoE Subscribers

In this topology:

- SN-19-RR is the Service Node which is a Linux container running on a Debian server. BNG Blaster, which emulates routing and access functions, runs on this container. This container also runs FreeRADIUS, which emulates RADIUS functions. BNG Blaster is an open-source network tester developed by RtBrick to validate the performance and scalability of BNG devices. The BNG blaster can emulate both subscribers at scale and network elements such as routers. Therefore, it is possible to create complex topologies with just the DUT and a server that hosts the BNG Blaster, thereby minimizing the equipment needed to test the DUT. For more

information, refer to the BNG Blaster documentation.

- There are three routers; two acting as core routers (R-1 & R-2) and the third one as the Edge router. R-1, R-2, and Edge Router are RBFS virtual helper modes. The devices need not be RtBrick Multiservice Edge.
- Multiservice Edge1 and Multiservice Edge2 are the DUTs, which are two bare metal switches installed with the RBFS Multiservice Edge software. On one side, these are connected to the Access Network, while on the other side, they are connected to the Core Network.
- The Multiservice Edge nodes establish IS-IS adjacencies and LDP sessions with core routers R1 and R2. The BNG-Blaster simulates four route reflectors (RRs) that have BGP sessions with the CBNGs, which are operating in redundant mode. Two of these emulated RRs advertise 1 million IPv4 prefixes each, and the other two advertise 250,000 IPv6 prefixes each. These prefixes are then advertised to the CBNGs through the BGP sessions.
- The topology brings up 20K IPoE subscribers (2 OLTs with 10K subscribers each simulated by BNG Blaster) over a LAG interface. A failure of a CBNG does not affect a subscriber's services since they are backed up between the CBNGs.

4.2. Configuration and Settings

4.2.1. Platform Configuration and Settings

This section provides information about the platform and how to set various required configurations for the platform.

Know your Device

The configurations provided in this reference design document (Multiservice Edge IPoE implementation) are generated on the UfiSpace S9600-72XC platform. 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 [1GbE/10GbE/25GbE] and 8x100GE [40GbE/100GbE] high-speed ports with a switching capacity of up to 2.4Tbs.

The RBFS Multiservice Edge software is installed on top of the UfiSpace S9600-72XC.



Although the specific device used here is UfiSpace S9600-72XC, the configuration will stay exactly the same for any other device that supports the Multiservice Edge image.

For more information about the hardware specifications of UfiSpace S9600-72XC, see the Platform Guide.

Prerequisites

- Access to BNG Blaster, an open-source network testing platform for access and routing protocols. For information on obtaining and building BNG Blaster, see https://rtbrick.github.io/bngblaster/.
- Access to FreeRADIUS, a free RADIUS suite. For accessing FreeRADIUS, see

https://freeradius.org/.

- Access to Syslog server.

Restore Configuration

Depending on the deployment scenario, a running configuration can be applied or restored as needed.

To enable configuration restore, enter the `set system load-last-config true` command as shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> set system load-last-config
true
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> commit
```

For more information, see the section "Running Configuration" of the RBFS NOC Troubleshooting Guide.

4.2.2. General Configuration

To enable testing some basic primitives need to be configured. These general configurations include loopback interface for identifying and accessing the device on network, NTP for setting accurate time across a whole network of devices, TACACS+ for user authentication, user management for user configuration, license for accessing RBFS, Resmon for resource monitoring, and Syslog configurations for exporting the log message to the external log management server.

Configure License

Without any license installed on your system, you can evaluate RBFS for 7 days. You need to get an evaluation license or purchase an actual license within 7 days to use the full functionality of RBFS.

The following steps provide the commands to install an RBFS license key. For more information about license configuration, see Installing License.

Switch to config mode using the switch-mode config command to continue with the RBFS configurations.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> switch-mode config
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg>
```

Install the license encrypted string (that is received from RtBrick) using the RBFS CLI.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> set system license <license-
key>
```

RBFS license configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show config system license
AAAAWsfg&jdkfs4D34H5@2evf...
```

As shown below, the "show system license" command displays the expiration date for the current license.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show system license
License Validity:
  License index 1:
    Start date : Tue Feb 28 09:44:27 GMT +0000 2023
    End date   : Mon Mar 04 09:44:27 GMT +0000 2024
supervisor@rtbrick>multiservice-edge.rtbrick.net: op>
```

Configure Instance

Instance `default` will be available by default without any configurations.

Create the instance `inband_mgmt` by entering the following commands.

```
set instance inband_mgmt
set instance inband_mgmt address-family ipv4 unicast
commit
```

The configurations of the instance `inband_mgmt` are shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config instance
inband_mgmt
{
  "rtbrick-config:instance": [
    {
      "name": "inband_mgmt",
      "address-family": [
        {
          "afi": "ipv4",
          "safi": "unicast"
        }
      ]
    }
  ]
}
```

Below are the configurations available for the available instances.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show instance detail
Instance: default
  Instance ID: 0
```

```
  State: Active
  AFI        SAFI             State
  ipv4       unicast          Active
  ipv4       multicast        Active
  ipv4       labeled-unicast  Active
  ipv6       unicast          Active
  ipv6       multicast        Active
  ipv6       labeled-unicast  Active
  mpls       unicast          Active
Instance: inband_mgmt
  Instance ID: 3
  State: Active
  AFI        SAFI             State
  ipv4       unicast          Active
```

Configure Loopback Interface

Loopback Interface configuration is required as it is the best way to identify a network device and is always reachable. Also, protocols use the loopback address to determine protocol-specific properties for the device.

The following steps provide the commands to configure the loopback interface. For more information about Loopback Interface configuration, see the Interfaces User Guide.

Configure loopback interface on the device.

```
set interface lo-0/0/1 unit 0 address ipv4 192.0.2.64/32
set interface lo-0/0/1 unit 1 address ipv4 192.0.2.74/32
set interface lo-0/0/1 unit 2 instance inband_mgmt
set interface lo-0/0/1 unit 2 address ipv4 192.0.2.128/32
set interface lo-0/0/1 unit 3 instance inband_mgmt
set interface lo-0/0/1 unit 3 address ipv4 192.0.2.131/32
commit
```



The configuration commands should be followed by the `commit` command to save the configurations into the device.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> commit
```

Loopback Interface configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config interface lo-0/0/1
{
  "rtbrick-config:interface": [
    {
      "name": "lo-0/0/1",
      "unit": [
```

```
        {
          "unit-id": 0,
          "address": {
            "ipv4": [
              {
                "prefix4": "192.0.2.64/32"
              }
            ]
          }
        },
        {
          "unit-id": 1,
          "address": {
            "ipv4": [
              {
                "prefix4": "192.0.2.74/32"
              }
            ]
          }
        },
        {
          "unit-id": 2,
          "instance": "inband_mgmt",
          "address": {
            "ipv4": [
              {
                "prefix4": "192.0.2.128/32"
              }
            ]
          }
        },
        {
          "unit-id": 3,
          "instance": "inband_mgmt",
          "address": {
            "ipv4": [
              {
                "prefix4": "192.0.2.131/32"
              }
            ]
          }
        }
      ]
    }
  ]
}
```

Configure IP Addresses for Core Interfaces

Enter the following commands to configure IP addresses for the core interfaces.

```
set interface ifp-0/1/31 unit 10
set interface ifp-0/1/31 unit 10 vlan 10
set interface ifp-0/1/31 unit 10 address ipv4 192.0.2.1/27
set interface ifp-0/1/31 unit 10 address ipv6 2001:db8::1/64
set interface ifp-0/1/31 unit 100
set interface ifp-0/1/31 unit 100 vlan 100
set interface ifp-0/1/31 unit 100 address ipv4 192.0.2.33/27
set interface ifp-0/1/31 unit 200
set interface ifp-0/1/31 unit 200 instance inband_mgmt
set interface ifp-0/1/31 unit 200 vlan 200
set interface ifp-0/1/31 unit 200 address ipv4 192.0.2.97/27
commit
```

Below configuration shows the IP address configurations for the core interfaces.

```
{
    "rtbrick-config:interface": [
      {
        "name": "ifp-0/1/31",
        "unit": [
          {
            "unit-id": 10,
            "vlan": 10,
            "address": {
              "ipv4": [
                {
                  "prefix4": "192.0.2.1/27"
                }
              ],
              "ipv6": [
                {
                  "prefix6": "2001:db8::1/64"
                }
              ]
            }
          },
          {
            "unit-id": 100,
            "vlan": 100,
            "address": {
              "ipv4": [
                {
                  "prefix4": "192.0.2.33/27"
                }
              ]
            }
          },
          {
            "unit-id": 200,
```

```
            "instance": "inband_mgmt",
            "vlan": 200,
            "address": {
              "ipv4": [
                {
                  "prefix4": "192.0.2.97/27"
                }
              ]
            }
          }
        ]
      }
    ]
  }
}
```

Configure Static Routes to Enable Reachability to the NTP and TACACS Servers

Below are static routes configured to enable reachability to the NTP (192.0.2.129) and TACACS (192.0.2.130) servers. On the Service Node, 192.0.2.98 is the interface address on VLAN 200. It is explained later in this document how to configure IP addresses on Service Node. For details, see Configuring Interfaces on the Service Node for NTP and TACACS Connectivity from Multiservice Edge.

```
set instance inband_mgmt static route ipv4 192.0.2.129/32 unicast np1
set instance inband_mgmt static route ipv4 192.0.2.130/32 unicast np1
set instance inband_mgmt static nexthop-profile np1
set instance inband_mgmt static nexthop-profile np1 nexthop 192.0.2.98
set instance inband_mgmt static nexthop-profile np1 lookup-instance inband_mgmt
set instance inband_mgmt static nexthop-profile np1 lookup-afi ipv4
set instance inband_mgmt static nexthop-profile np1 lookup-safi unicast
commit
```

The configuration of the static routes is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config instance
inband_mgmt static
{
  "rtbrick-config:static": {
    "route": {
      "ipv4": [
        {
          "prefix4": "192.0.2.129/32",
          "safi": "unicast",
          "nexthop-profile": "np1"
        },
        {
          "prefix4": "192.0.2.130/32",
```

```
          "safi": "unicast",
          "nexthop-profile": "np1"
        }
      ]
    },
    "nexthop-profile": [
      {
        "name": "np1",
        "nexthop": "192.0.2.98",
        "lookup-instance": "inband_mgmt",
        "lookup-afi": "ipv4",
        "lookup-safi": "unicast"
      }
    ]
  }
}
```

Configure NTP

Configuring NTP (Network Time Protocol) provides time synchronization across a whole network of devices. An NTP network consists devices (clients) which are to be synchronized with the NTP server that provides accurate time to the client devices.

The following steps provide the commands to configure Network Time Protocol (NTP) for the device. For more information about NTP configuration, see the NTP User Guide.

Enabling NTP Service:

To access the NTP service running in the RtBrick Host, this service has to be enabled in inband-management. On configuring this, the hosts reachable in inband instance via the physical interface can access this service.

Configure NTP server and NTP service on the device.

```
set system ntp server ntp1
set system ntp server ntp1 ipv4-address 192.0.2.129
set inband-management instance inband_mgmt
set inband-management instance inband_mgmt ntp true
commit
```

NTP configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config inband-management
{
    "rtbrick-config:inband-management": {
      "instance": [
        {
          "name": "inband_mgmt",
          "ntp": "true"
```

```
        }
      ]
    }
  }
```

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config system ntp
{
    "rtbrick-config:ntp": {
      "server": [
        {
          "server-name": "ntp1",
          "ipv4-address": "192.0.2.129"
        }
      ]
    }
  }
```

User Authentication

RBFS supports user authentication through a centralized TACACS+ server and with a local authentication system. The following authentication process typically occurs when a user attempts to access the network.

1. When a user logs in through SSH, the SSH Daemon (sshd) invokes the Pluggable Authentication Module (PAM) to trigger authentication process.
2. PAM requests TACACS+ authentication (except for the user with the supervisor privileges).
3. TACACS+ server provides 'grant access' node if the user authentication is successful.
4. If the user is not allowed using the TACACS+ authentication, it is required to undergo an additional authentication phase. PAM looks up local users. Upon successful authentication, PAM generates RTB PAM token; includes user role in 'scope'.

Define Users on TACACS+ Server

Administrator needs to define users and associate them with the predefined roles on the TACACS+ server. Optionally, RBFS CLI commands can be restricted using the rtb-allow-cmds and rtb-deny-cmds.

The tac_plus.conf file contains configuration information for the tac_plus(tacacs+) daemon. This file is stored at the following location:

/etc/tacacs+/tac_plus.conf

To view the TACACS+ server configuration file, enter the following command.

```
sudo cat /etc/tacacs+/tac_plus.conf
```

For more information about TACACS+ server configuration, see

https://manpages.ubuntu.com/manpages/trusty/man5/tac_plus.conf.5.html

This Reference Design document uses the default local user `supervisor` for the configurations, whereas other users, defined in the TACACS server, can log into RBFS by using their usernames and passwords.

The following TACACS+ configuration shows the details of the TACACS users.

- Click here to download the `tac_plus.conf` file.

Configure TACACS+ on RBFS

After defining the users on the TACACS+ server, configure the TACACS+ server on Multiservice Edge. This configuration allows the remote TACACS+ server to communicate with the Multiservice Edge and to validate user access on the network.

The following steps provide the commands to configure TACACS+. For more information about TACACS+ configuration, see the Configure TACACS+ on RBFS.

To access the TACACS+ service running in the RtBrick Host, this service has to be enabled in inband management. On configuring this, the hosts reachable in inband instance via the physical interface can access this service.

```
set system secure-management-status true
set system authorization tacacs 192.0.2.130 inband secret-plain-text
RtBrick_Little_Secret
set inband-management instance inband_mgmt tacacs true
commit
```

In the above configuration, the command `set inband-management instance inband_mgmt tacacs true` is used to enable TACACS+ under the instance called `inband_mgmt`.

TACACS+ configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config system
authorization
{
  "rtbrick-config:authorization": {
    "tacacs": [
      {
        "ipv4-address": "192.0.2.130",
        "type": "inband",
        "secret-encrypted-text":
"$22464b2c7336cfe71e596c447be28d598b9b7b37f92faea157fd5058e5fe0d769"
      }
    ]
  }
}
```

Configuration for enabling TACACS+ under the instance `inband_mgmt` is shown below:

```
"rtbrick-config:inband-management": {
      "instance": [
        {
          "name": "inband_mgmt",
          "tacacs": "true"
        }
      ]
    },
```

Enabling TACACS+ Service on the Service Node

Enter the following commands to enable the TACACS service on the Service Node.

```
~$ sudo /bin/systemctl enable tacacs_plus.service
~$ sudo /bin/systemctl start tacacs_plus.service
```

Validating TACACS+ authentication

The following scenario shows a successful authentication for the user `bob` with password `bob`.

```
~$ ssh bob@multiservice-edge.rtbrick.net
bob@multiservice-edge.rtbrick.net's password:
Last login: Mon Apr 3 16:13:40 2023 from multiservice-edge.rtbrick.net
bob@rtbrick>multiservice-edge.rtbrick.net: op>
```

The following scenario shows an unsuccessful password authentication for the user `bob` with password `bob123`.

```
~$ ssh bob@rtbrick>multiservice-edge.rtbrick.net:
bob@multiservice-edge.rtbrick.net's password:
Permission denied, please try again.
bob@multiservice-edge.rtbrick.net's password:
```

The following scenario shows an unsuccessful authentication for an undefined user `frank`.

```
~$ ssh frank@rtbrick>multiservice-edge.rtbrick.net:
frank@multiservice-edge.rtbrick.net's password:
Permission denied, please try again.
frank@multiservice-edge.rtbrick.net's password:
accounting file = /var/log/tac_plus.acct
key = RtBrick_Little_Secret
```

Configure User Management

Configuring Local User Management enables administrators to create, manage, and secure the users and groups. It allows creation of privileges that are configurable for user-defined and predefined roles.

The following steps provide the commands to configure user management. For more information about license configuration, see the Local User Management.

1. To create a role, configure the RBAC privilege and the command privilege. To configure the RBAC privilege for both table and object:

```
set system user admin role supervisor
set system user admin shell /bin/bash
set system user admin password-hashed-text
$6$XNkmuMRI.5.R/NBJ$XDfZec7gEM3z/3lYn8mDDWimRZ/68xawia.pTMdrGqoYHEE3nWHB08DeaPNQTwHW6W
jB1aX6.xjYjh8CNCy4g1
commit
```

For information about Configuring hashed password, see Configure Hashed Password.

Authentication configuration of a password hashed text and an SSH public key is shown below:

```
{
  "ietf-restconf:data": {
    "rtbrick-config:system": {
      "user": [
        {
          "username": "admin",
          "shell": "/usr/local/bin/cli",
          "password-hashed-text":
"$5$L2DaOYYuddhBV$9RA5MX9RQzLC9fIKJzbnoFBb88w9rkSXl7GVrVJ9PY7",
          "ssh-pub-key": [
            "ssh-rsa AAAAWsfg&jdkfs4D34H5@2evf....."
            ]
        }
      ]
    }
  }
}
```

Configure Syslog

RBFS supports sending log messages to a Syslog server. The Syslog configuration can be performed in RBFS.

To configure logging for `bgp` by using Syslog, enter the following commands.

```
set log module bgp
set log module bgp level debug
set log module bgp plugin-alias
set log module bgp plugin-alias alias-name syslog
set log module bgp plugin-alias level debug
commit
```



For event logging, CtrlD only supports Graylog and Syslog. Graylog must be disabled in order to enable Syslog. In addition, Graylog attributes must be replaced with Syslog attributes.

Sylog configuration for the module `bgp` is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config log
{
  "rtbrick-config:log": {
    "module": [
      {
        "module-name": "bgp",
        "level": "debug",
        "plugin-alias": {
          "alias-name": "syslog",
          "level": "debug"
        }
      }
    ]
  }
}
```

Accessing the RtBrick Host to Configure Syslog

The steps described in this section are performed on the RtBrick Host. For logging into the RtBrick Host, use SSH port 1022.

```
ssh supervisor@<multiservice-edge-management-ip> -p 1022
```

After logging into the RtBrick Host, go to the following location of CtrlD and edit the `config.json` file.

- `/etc/rtbrick/ctrld/config.json`

Specify the Syslog configurations as shown below in the `config.json` file.

```
{
    "rbms_enable": false,
    "graylog_enable": false,
    "syslog_enable": true,
```

```
    "syslog_network": "udp",
    "syslog_urls": [
      "198.51.100.49:516"
    ],
    "syslog_severity_level": 7,
    "auth_enabled": false
}
```



- For documentation purposes, the IP address `198.51.100.49` has been used as the IP address of the Syslog endpoint. This IP address should be updated with the actual Syslog server's IP address.
- Syslog messages can be transported using UDP or TCP protocol. In this configuration, Syslog messages are transported using `udp`.

After making configuration changes in the `config.json`, restart CtrlD service as shown below.

```
supervisor@onl>multiservice-edge.rtbrick.net:~ $ sudo service rtbrick-ctrld restart
[sudo] password for supervisor:
[ ok ] Stopping rtbrick ctrld service:.
[ ok ] Starting rtbrick ctrld service:.
```

Monitor Resources (Resmon)

Resource monitoring enables administrators to collect and analyze the health information and usage data of various hardware resources such as CPU, memory, processes, disks, sensors, optics, and so on.

Run `show cpu usage`, `show memory usage` and `show disk usage` to see the CPU, memory and disk utilization respectively.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show cpu usage
Name          Total       User     System       Nice   I/O Wait       Idle        IRQ
Soft IRQ
cpu             11%         9%         1%         0%         0%        88%         0%
      0%
cpu0            10%         8%         1%         0%         0%        89%         0%
      0%
cpu1            44%        43%         1%         0%         0%        55%         0%
      0%
cpu2            34%        32%         2%         0%         0%        66%         0%
      0%
cpu3            11%         9%         2%         0%         0%        89%         0%
      0%
cpu4             3%         1%         2%         0%         0%        96%         0%
      0%
cpu5             4%         3%         1%         0%         0%        96%         0%
      0%
```

```
cpu6            13%         11%          2%          0%          0%         87%          0%
      0%
cpu7            24%         22%          1%          0%          0%         75%          0%
      0%
cpu8             4%          2%          2%          0%          0%         95%          0%
      0%
cpu9             2%          1%          1%          0%          0%         97%          0%
      0%
cpu10            6%          4%          2%          0%          0%         93%          0%
      0%
cpu11            3%          2%          1%          0%          0%         96%          0%
      0%
cpu12            8%          6%          2%          0%          0%         91%          0%
      0%
cpu13            3%          3%          0%          0%          0%         96%          0%
      0%
cpu14            6%          4%          2%          0%          0%         93%          0%
      0%
cpu15            8%          5%          2%          0%          0%         91%          0%
      0%
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show memory usage
Name     Total            Used             Free          Shared       Buffers
 Cached
RAM      31.03 GiB        8.51 GiB         17.04 GiB    1.19 GiB     112.66 MiB   5.37
GiB
SWAP     0 bytes          0 bytes          0 bytes      n/a          n/a          n/a
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show disk usage
Filesystem               Type             Size         Used         Available
 Mountpoint          Usage %
none                     tmpfs            492 KiB      0 bytes      492 KiB      /dev
               0.0
tmpfs                    tmpfs            15.51 GiB    17.23 MiB    15.5 GiB     /run
               0.11
tmpfs                    tmpfs            6 GiB        828.75 MiB   5.19 GiB     /shm
               13.49
tmpfs                    tmpfs            15.51 GiB    182.96 MiB   15.33 GiB
 /dev/shm            1.15
tmpfs                    tmpfs            5 MiB        0 bytes      5 MiB
 /run/lock           0.0
devtmpfs                 devtmpfs         1 MiB        0 bytes      1 MiB
 /dev/mem            0.0
/dev/sda10               ext4             15.62 GiB    50.61 MiB    14.76 GiB
 /var/log            0.33
/dev/sda6                ext4             29.4 GiB     4.24 GiB     23.65 GiB
 /platform           15.2
tmpfs                    tmpfs            3.1 GiB      0 bytes      3.1 GiB
 /run/user/1000      0.0
tmpfs                    tmpfs            3.1 GiB      0 bytes      3.1 GiB
 /run/user/1001      0.0
tmpfs                    tmpfs            15.51 GiB    0 bytes      15.51 GiB
 /sys/fs/cgroup      0.0
```

```
/dev/sda11               ext4            43.79 GiB    51.89 MiB    41.49 GiB
 /var/crash          0.12
tmpfs                    tmpfs           3.1 GiB      1.02 MiB     3.1 GiB
 /var/run-ext/host/r  0.03
/var/cache/rtbrick/imag  overlay         29.4 GiB     4.24 GiB     23.65 GiB    /
              15.2
```

The show command can also be used to view other resource details. For information about the resmon configuration and operational commands, see the RBFS Resource Monitoring Guide.

4.3. Protocol Configurations

This validated solution design topology uses IS-IS as the interior gateway protocol to distribute IP routing information among the routers in an Autonomous System (AS). The Label Distribution Protocol (LDP) is used to exchange label mapping information for MPLS traffic. And, iBGP is used for exchanging routing and reachability information within ASs.

One thus needs to configure the following protocols:

- IS-IS : To ensure IP connectivity on the core network.
- LDP : To establish MPLS LSP tunnels for MPLS data transmission on the network.
- iBGP: To exchange routing information within an AS.

4.3.1. Configure IS-IS

The following steps provide the commands to execute various IS-IS protocol functionalities. For more detailed information about IS-IS configuration, see the IS-IS User Guide.

1. Configure IS-IS system-id, area, hostname and interfaces.

```
set instance default protocol isis system-id 1920.0000.2064
set instance default protocol isis area 49.0001/24
set instance default protocol isis hostname multiservice-edge
set instance default protocol isis interface ifl-0/1/31/10
set instance default protocol isis interface ifl-0/1/31/10 type point-to-point
set instance default protocol isis interface ifl-0/1/31/10 level-2 adjacency-disable
true
set instance default protocol isis interface lo-0/0/1/0
set instance default protocol isis interface lo-0/0/1/0 passive true
commit
```

IS-IS instance configuration on interface is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show config instance default
protocol isis
{
  "rtbrick-config:isis": {
```

```
    "system-id": "1920.0000.2064",
    "area": [
      "49.0001/24"
      ],
    "hostname": "multiservice-edge",
    "interface": [
      {
        "name": "ifl-0/1/31/10",
        "type": "point-to-point",
        "level-2": {
          "adjacency-disable": "true"
        }
      },
      {
        "name": "lo-0/0/1/0",
        "passive": "true"
      }
    ]
  }
}
```

4.3.2. Configure LDP on the Interfaces

The following steps provide the commands to execute various LDP functionalities. For more detailed information about LDP configuration, see *LDP User Guide*.

1. Configure LDP on the router interface.

```
set instance default protocol ldp router-id 192.0.2.64
set instance default protocol ldp interface ifl-0/1/31/10
set instance default protocol ldp interface lo-0/0/1/0
commit
```

Configuration for LDP on the interface is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config instance default
protocol ldp
{
  "rtbrick-config:ldp": {
    "router-id": "192.0.2.64",
    "interface": [
      {
        "name": "ifl-0/1/31/10"
      },
      {
        "name": "lo-0/0/1/0"
      }
    ]
```

```
  }
}
```

4.3.3. Configure BGP

The following steps provide the commands to execute the various BGP functionalities quickly. For more detailed information about BGP configuration, see the BGP User Guide.

1. Configure BGP local AS, router-id, and hostname

```
set instance default protocol bgp local-as 4200000001
set instance default protocol bgp router-id 192.0.2.64
set instance default protocol bgp hostname multiservice-edge
commit
```

BGP local AS, router-id, and hostname configurations are shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config instance default
protocol bgp
{
  "rtbrick-config:bgp": {
    "local-as": 4200000001,
    "hostname": "multiservice-edge",
    "router-id": "192.0.2.64",

<....>
```

2. Enable the IPv4 and IPv6 address families which are to be supported on the specific BGP instance.

```
set instance default protocol bgp address-family ipv4 unicast
set instance default protocol bgp address-family ipv4 unicast resolve-nexthop safi
labeled-unicast
set instance default protocol bgp address-family ipv6 labeled-unicast
set instance default protocol bgp address-family ipv6 unicast
set instance default protocol bgp address-family ipv6 unicast resolve-nexthop safi
labeled-unicast
commit
```

BGP address family configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config instance default
protocol bgp address-family
{
  "rtbrick-config:address-family": [
    {
```

```
      "afi": "ipv4",
      "safi": "unicast",
      "resolve-nexthop": {
        "safi": "labeled-unicast"
      }
    },
    {
      "afi": "ipv6",
      "safi": "labeled-unicast"
    },
    {
      "afi": "ipv6",
      "safi": "unicast",
      "resolve-nexthop": {
        "safi": "labeled-unicast"
      }
    }
  ]
}
```

3. Create the peer group with the specific remote AS configurations and the address family that is to be negotiated with the peer which will be attached to the peer group later.

```
set instance default protocol bgp peer-group RR
set instance default protocol bgp peer-group RR remote-as 4200000001
set instance default protocol bgp peer-group RR address-family ipv4 unicast
set instance default protocol bgp peer-group RR address-family ipv6 labeled-unicast
set instance default protocol bgp peer-group RR address-family ipv6 unicast
commit
```

BGP peer-group configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config instance default
protocol bgp peer-group
{
  "rtbrick-config:peer-group": [
    {
      "pg-name": "RR",
      "remote-as": 4200000001,
      "address-family": [
        {
          "afi": "ipv4",
          "safi": "unicast"
        },
        {
          "afi": "ipv6",
          "safi": "labeled-unicast"
        },
        {
```

```
          "afi": "ipv6",
          "safi": "unicast"        }
      ]
    }
  ]
}
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg>
```

4. Add a BGP peer and associate it with the specific peer group.

```
set instance default protocol bgp peer
set instance default protocol bgp peer ipv4 198.51.100.3 192.0.2.64
set instance default protocol bgp peer ipv4 198.51.100.3 192.0.2.64 peer-group RR
commit
```

Configuration for adding a BGP peer and associating it with a peer group is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config instance default
protocol bgp peer
{
  "rtbrick-config:peer": {
    "ipv4": [
      {
        "peer-address": "198.51.100.3",
        "update-source": "192.0.2.64",
        "peer-group": "RR"
      }
    ]
  }
}
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg>
```

5. Configure the IPv6 unicast address family with `send-label` as true, then address-family IPv6 labeled-unicast gets negotiated with the peer.

```
set instance default protocol bgp peer-group RR address-family ipv6 unicast send-label
true
commit
```

The following configuration shows the BGP IPv6 unicast address family with `send-label` as true.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config instance default
protocol bgp peer-group RR address-family ipv6 unicast
{
  "rtbrick-config:address-family": [
    {
```

```
      "afi": "ipv6",
      "safi": "unicast",
      "send-label": "true"
    }
  ]
}
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg>
```

6. Set the resolve-nexthop, if the BGP nexthop attribute of the BGP routes needs to be resolved under ipv4/ipv6 labeled-unicast routing table. It configures only resolve-nexthop safi. Based on the nexthop-type (ipv4 or ipv6), it gets looked up into either IPv4 labeled-unicast or IPv6 labeled-unicast.

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

Resolve nexthop configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config instance default
protocol bgp address-family ipv4 unicast resolve-nexthop
{
  "rtbrick-config:resolve-nexthop": {
    "safi": "labeled-unicast"
  }
}
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg>
```

7. To redistribute the routes (belonging to a specific source) into BGP, execute the following command. The following command redistributes `direct` routes into BGP.

```
set instance default protocol bgp address-family ipv4 unicast redistribute direct
commit
```

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config instance default
protocol bgp address-family ipv4 unicast redistribute
{
  "rtbrick-config:redistribute": [
    {
      "source": "direct"
    }
  ]
}
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg>
```

4.3.4. Configuring the Service Node

Configuring Interfaces on the Service Node for RADIUS Connectivity from Multiservice Edge

To configure interfaces on the Service Node for RADIUS connectivity from Multiservice Edge, enter the following commands:

```
sudo ip link add link SN1-3-C1 name SN1-3-C1.100 type vlan id 100
sudo ifconfig SN1-3-C1.100 192.0.2.34/27
```



`SN1-3-C1` is the internal nomenclature that denotes the interface name on Service Node that connects to the Multiservice Edge.

Configuring Interfaces on the Service Node for NTP and TACACS Connectivity from Multiservice Edge

To configure interfaces on the Service Node for NTP and TACACS connectivity from Multiservice Edge, enter the following commands:

```
sudo ip link add link SN1-3-C1 name SN1-3-C1.200 type vlan id 200
sudo ifconfig SN1-3-C1.200 192.0.2.98/27
sudo ifconfig lo:1 192.0.2.129 netmask 255.255.255.255 up
sudo ifconfig lo:2 192.0.2.130 netmask 255.255.255.255 up
```

Configuring Routes on the Service Node

To configure routes on the Service Node that provides reachability to the RADIUS, TACACS and NTP servers, enter the following commands:

```
sudo ip route add 192.0.2.74/32 via 192.0.2.33
sudo ip route add 192.0.2.131/32 via 192.0.2.97
sudo ip route add 192.0.2.128/32 via 192.0.2.97
```

BNG Blaster Configuration for Protocols

BNG Blaster is an open-source network testing platform for access and routing protocols. It can emulate massive PPPoE and IPoE (DHCP) subscribers including IPTV, and L2TP (LNS). There are various routing protocols supported such as IS-IS and BGP. So, one can use this platform for end-to-end BNG and non-BNG router testing.

For more information about BNG Blaster, see https://github.com/rtbrick/bngblaster

For information about installing BNG Blaster, see https://rtbrick.github.io/bngblaster/install.html

Downloading the Blaster Configuration File

The following is the configuration file that is used in BNG Blaster for validating IPoE, BGP, IS-IS, and

LDP.

- Click here to download the BNG Blaster configuration file (`blaster.json`).

Generating Supporting Files for Protocols

- **Generating BGP Internet Prefixes**

Enter the following commands to generate BGP internet prefixes.

```
bgpupdate -f internet.bgp -a 4200000001 -n 198.51.100.2 -N 1 -p 172.16.0.0/24 -P
1000000
bgpupdate -f internet.bgp -a 4200000001 -n 198.51.100.2 -N 1 -p 2004::/48 -m 10000 -M
5 -P 150000 --append
```

Ensure that the command execution has finished (as shown below) before continuing.

```
[2023-04-05 10:19:32][INFO   ] init 1000000 IPv4 prefixes
[2023-04-05 10:19:56][INFO   ] open file internet.bgp (replace)
[2023-04-05 10:25:31][INFO   ] finished
```

After the generation of the `internet.bgp` file, the "raw-update-file" attribute of the `blaster.json` file needs to be updated as follows:

```
"raw-update-file": "/home/supervisor/internet.bgp"
```



For more information about downloading the blaster.json file, see section Downloading the Blaster Configuration File.

- **Generating MRT FIle for IS-IS**

Below is the JSON file (`isis_3node.json`) which is used to simulate R-1, R-2, and RR on BNG Blaster.

- Click here to download the ` isis_3node.json ` file.

This JSON file needs to converted to MRT format using the following command:

```
lspgen -r isis_3node.json -m isis.mrt
```

After converting the file to `isis.mrt`, it needs to be updated in the IS-IS section in the `blaster.json` file as shown below.

```
"mrt-file": "/home/supervisor/isis.mrt"
```

- Generate labels for the IS-IS prefixes using "ldpupdate" command as shown below:

```
ldpupdate -l 192.0.2.65 -p 198.51.100.1/32 -P 3 -M 3 -f out.ldp
```

The detail of the generated file needs to be added to the LDP section in the `blaster.json` file as shown below:

```
"raw-update-file": "/home/supervisor/out.ldp"
```

Starting BNG Blaster

In the following command line string, a BNG Blaster instance is started and the `blaster.json` file is used.

```
sudo bngblaster -C blaster.json -I
```

The **-C blaster.json** argument specifies the blaster configuration file. The -I flag enables the interactive blaster UI.

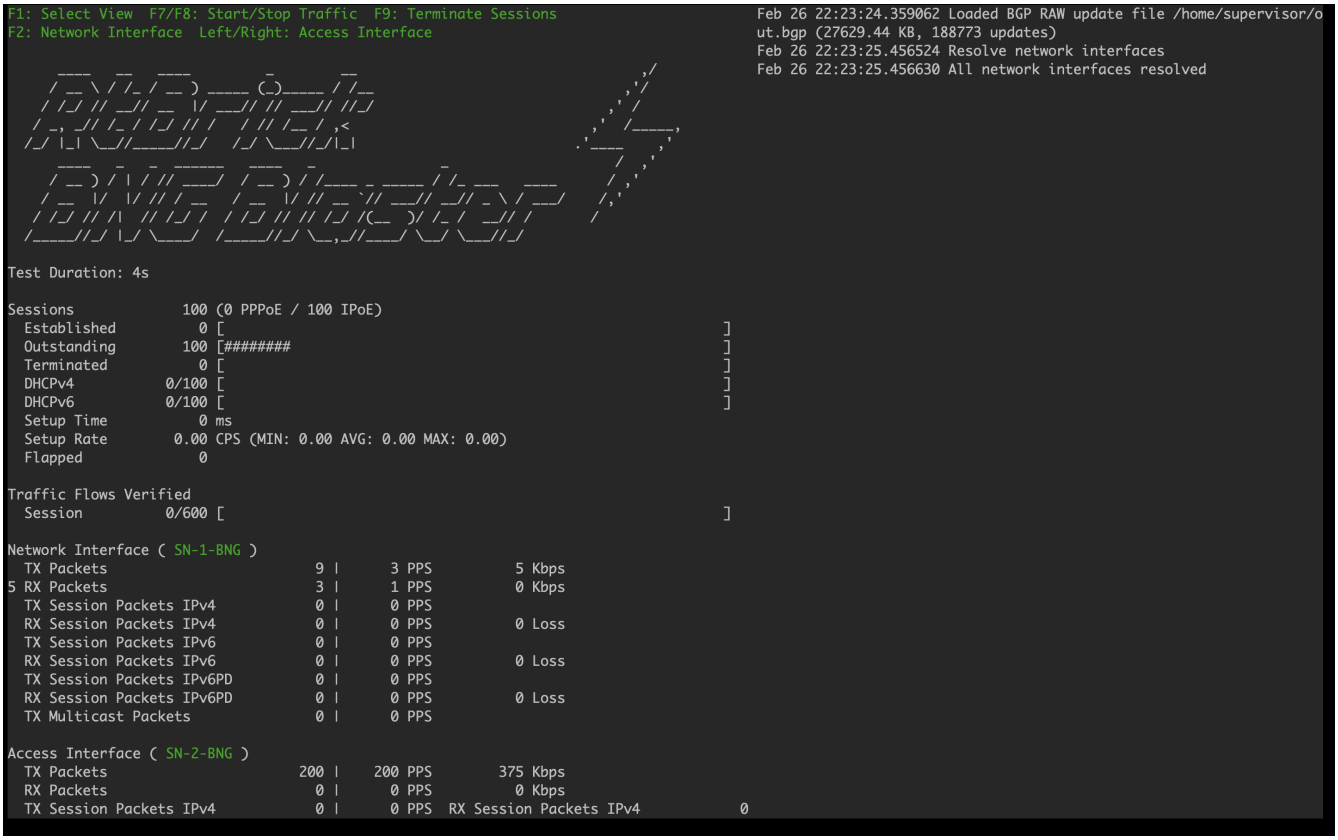


Figure 15. Starting BNG Blaster

4.3.5. Validating Protocols on RBFS Multiservice Edge

Validating IS-IS Adjacency, Routes and Reachability

Run the following command to show IS-IS adjacency.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show isis neighbor
Instance: default
  Interface          System             Level   State   Type    Up since               Expires
  ifl-0/1/31/10       BNG-Blaster        L1      Up      P2P     Wed Mar 29 05:23:24     in 22s 242106us
```

After configuring IS-IS protocol, check the IPv4 unicast routes, populated by IS-IS using the following command:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show route ipv4 unicast source isis instance default
Instance: default, AFI: ipv4, SAFI: unicast
Prefix/Label                          Source           Pref    Next Hop                              Interface

192.0.2.2/27                          isis             15      192.0.2.2                             ifl-0/1/31/10
192.0.2.65/32                         isis             15      192.0.2.2                             ifl-0/1/31/10
198.51.100.1/32                       isis             15      192.0.2.2                             ifl-0/1/31/10
198.51.100.2/32                       isis             15      192.0.2.2                             ifl-0/1/31/10
198.51.100.3/32                       isis             15      192.0.2.2                             ifl-0/1/31/10
198.51.100.101/31                     isis             15      192.0.2.2                             ifl-0/1/31/10
198.51.100.103/31                     isis             15      192.0.2.2                             ifl-0/1/31/10
198.51.100.105/31                     isis             15      192.0.2.2                             ifl-0/1/31/10
```

Ping the address 192.0.2.65 as follows:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> ping 192.0.2.65
68 bytes from 192.0.2.65: icmp_seq=1 ttl=64 time=9.0436 ms
68 bytes from 192.0.2.65: icmp_seq=2 ttl=64 time=2.0959 ms
68 bytes from 192.0.2.65: icmp_seq=3 ttl=64 time=4.7229 ms
68 bytes from 192.0.2.65: icmp_seq=4 ttl=64 time=9.2496 ms
68 bytes from 192.0.2.65: icmp_seq=5 ttl=64 time=2.6149 ms
Statistics: 5 sent, 5 received, 0% packet loss
```

Validating LDP Adjacency, Routes and Reachability

Run the following commands to show LDP neighbor and LDP session.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show ldp neighbor
Instance: default
  Interface           LDP ID              Transport IP  Up Since                  Expires
  ifl-0/1/31/10        192.0.2.65:0        192.0.2.65    Wed Mar 29 05:21:11       in 11s
```

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show ldp session
Instance: default
  LDP ID            Peer IP           State             Up/Down           FECRcvd   FECSent
  192.0.2.65:0      192.0.2.65        Operational       0d:03h:55m:49s          5         5
```

After configuring the LDP protocol, check the IPv4 labeled unicast routes, populated by LDP using the following command:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op>  show route ipv4 labeled-unicast source ldp
Instance: default, AFI: ipv4, SAFI: labeled-unicast
Prefix/Label                            Source           Pref    Next Hop                              Interface
Label
192.0.2.2/27                            ldp              9       192.0.2.2                             ifl-0/1/31/10
-
192.0.2.65/32                           ldp              9       192.0.2.2                             ifl-0/1/31/10
-
198.51.100.1/32                         ldp              9       192.0.2.2                             ifl-0/1/31/10
10000
198.51.100.2/32                         ldp              9       192.0.2.2                             ifl-0/1/31/10
10001
198.51.100.3/32                         ldp              9       192.0.2.2                             ifl-0/1/31/10
10002
```

Ping the labeled unicast address 198.51.100.1 as follows:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> ping 198.51.100.1 instance
default afi ipv4 safi labeled-unicast
68 bytes from 198.51.100.1: icmp_seq=1 ttl=64 time=6.3289 ms
68 bytes from 198.51.100.1: icmp_seq=2 ttl=64 time=2.8249 ms
68 bytes from 198.51.100.1: icmp_seq=3 ttl=64 time=1.8587 ms
68 bytes from 198.51.100.1: icmp_seq=4 ttl=64 time=5.9599 ms
68 bytes from 198.51.100.1: icmp_seq=5 ttl=64 time=4.3811 ms
Statistics: 5 sent, 5 received, 0% packet loss
```



The command argument `labeled-unicast` takes the ICMP requests through a labeled path while validating IP connectivity and hence, it prepends an MPLS label.

Validating BGP Adjacency, Routes and Reachability

Run the following commands to show BGP session and state.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show bgp peer
Instance: default
  Peer                                  Remote AS    State         Up/Down Time              PfxRcvd
PfxSent
  198.51.100.3                          4200000001   Established   0d:00h:01m:24s            1150000             5
```

After configuring BGP, check the IPv4 unicast routes, populated by BGP using the following command:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show route ipv4 unicast source bgp instance default
Instance: default, AFI: ipv4, SAFI: unicast
Prefix/Label                            Source           Pref    Next Hop                              Interface
172.16.0.0/24                           bgp              200     198.51.100.2                          ifl-0/1/31/10
172.16.1.0/24                           bgp              200     198.51.100.2                          ifl-0/1/31/10
172.16.2.0/24                           bgp              200     198.51.100.2                          ifl-0/1/31/10
172.16.3.0/24                           bgp              200     198.51.100.2                          ifl-0/1/31/10
172.16.4.0/24                           bgp              200     198.51.100.2                          ifl-0/1/31/10
172.16.5.0/24                           bgp              200     198.51.100.2                          ifl-0/1/31/10
172.16.6.0/24                           bgp              200     198.51.100.2                          ifl-0/1/31/10
```

```
172.16.7.0/24                   bgp             200     198.51.100.2                    ifl-0/1/31/10
172.16.8.0/24                   bgp             200     198.51.100.2                    ifl-0/1/31/10
172.16.9.0/24                   bgp             200     198.51.100.2                    ifl-0/1/31/10

<...>
```

This command will list all the 1 million IPv4 BGP internet prefixes.

Pinging an IPv4 route (source: bgp) from the Multiservice Edge.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> ping 172.16.1.0
68 bytes from 172.16.1.0: icmp_seq=1 ttl=64 time=6.0527 ms
68 bytes from 172.16.1.0: icmp_seq=2 ttl=64 time=6.2893 ms
68 bytes from 172.16.1.0: icmp_seq=3 ttl=64 time=2.5573 ms
68 bytes from 172.16.1.0: icmp_seq=4 ttl=64 time=4.6964 ms
68 bytes from 172.16.1.0: icmp_seq=5 ttl=64 time=5.6455 ms
Statistics: 5 sent, 5 received, 0% packet loss
```

Check the IPv6 unicast routes, populated by BGP using the following command:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show route ipv6 unicast source
bgp instance default
Instance: default, AFI: ipv6, SAFI: unicast
Prefix/Label                            Source          Pref    Next Hop
Interface
2004::/48                               bgp             200     198.51.100.2
ifl-0/1/31/10
2004:0:1::/48                           bgp             200     198.51.100.2
ifl-0/1/31/10
2004:0:2::/48                           bgp             200     198.51.100.2
ifl-0/1/31/10
2004:0:3::/48                           bgp             200     198.51.100.2
ifl-0/1/31/10
2004:0:4::/48                           bgp             200     198.51.100.2
ifl-0/1/31/10
2004:0:5::/48                           bgp             200     198.51.100.2
ifl-0/1/31/10
2004:0:6::/48                           bgp             200     198.51.100.2
ifl-0/1/31/10
2004:0:7::/48                           bgp             200     198.51.100.2
ifl-0/1/31/10
2004:0:8::/48                           bgp             200     198.51.100.2
ifl-0/1/31/10
2004:0:9::/48                           bgp             200     198.51.100.2
ifl-0/1/31/10
2004:0:a::/48                           bgp             200     198.51.100.2
ifl-0/1/31/10
2004:0:b::/48                           bgp             200     198.51.100.2
ifl-0/1/31/10
2004:0:c::/48                           bgp             200     198.51.100.2
```

```
ifl-0/1/31/10
2004:0:d::/48                           bgp             200     198.51.100.2
ifl-0/1/31/10
2004:0:e::/48                           bgp             200     198.51.100.2
ifl-0/1/31/10
2004:0:f::/48                           bgp             200     198.51.100.2
ifl-0/1/31/10

<...>
```

Pinging an IPv6 route (source: bgp) from the Multiservice Edge.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> ping 2004:0:1::
68 bytes from 2004:0:1::: icmp_seq=1 ttl=253 time=10.0398 ms
68 bytes from 2004:0:1::: icmp_seq=2 ttl=253 time=2.9673 ms
68 bytes from 2004:0:1::: icmp_seq=3 ttl=253 time=6.2365 ms
68 bytes from 2004:0:1::: icmp_seq=4 ttl=253 time=7.9022 ms
68 bytes from 2004:0:1::: icmp_seq=5 ttl=253 time=1.5511 ms
Statistics: 5 sent, 5 received, 0% packet loss
```

4.4. IPoE Subscriber Management Configuration

IP-over-Ethernet (IPoE) is an access technology that uses DHCP for IPv4 and DHCPv6 for IPv6 where both protocols are handled in the IPoE daemon (`ipoed`). IPoE subscribers are identified by IFP, VLANs and client MAC addresses.

The dynamic creation of IPoE subscribers is triggered by DHCPv4 discover or DHCPv6 solicit request from the subscriber. Response is postponed until the subscriber is successfully authenticated using the known authentication methods such as local or RADIUS, however authentication is not mandatory. After the authentication phase, IPv4/IPv6/IPv6-PD address is allocated to the subscriber either from the local pool or from RADIUS.

For IPoE Subscriber Management, the following configurations are mandatory:

1. Access Interface Configuration
2. Access Profile Configuration
3. AAA (Authentication, Authorization and Accounting) Profile Configuration. Based on the authentication requirement, configure any one of the following:
    a. Local Authentication
        i. Pool Configuration
        ii. User Profile Configuration
    b. RADIUS Authentication
        i. RADIUS Profile Configuration
        ii. RADIUS Server Configuration

This solution section discusses RADIUS authentication.

NOTES:

- Access interfaces can be configured without VLAN tags (untagged) and with one (single tagged) or two (double tagged) VLAN tags.
- There can be more than one interface configured for subscriber management and each interface can reference the same profile.

4.4.1. Configuring IPoE Subscriber Management

For detailed information about the subscriber configuration options, see the Subscriber Management Configuration Guide.

1. Configure the access profile `ipoe`.

```
set access access-profile ipoe
set access access-profile ipoe protocol dhcp enable true
set access access-profile ipoe protocol dhcp lease-time 60
set access access-profile ipoe protocol dhcpv6 enable true
set access access-profile ipoe protocol dhcpv6 lifetime 60
set access access-profile ipoe address-family ipv4 enable true
set access access-profile ipoe address-family ipv4 pool-name poolv4
set access access-profile ipoe address-family ipv4 instance default
set access access-profile ipoe address-family ipv4 dad-enable true
set access access-profile ipoe address-family ipv6 enable true
set access access-profile ipoe address-family ipv6 pool-name poolv6
set access access-profile ipoe address-family ipv6 prefix-delegation-pool-name
poolv6pd
set access access-profile ipoe address-family ipv6 instance default
set access access-profile ipoe address-family ipv6 dad-enable true
commit
```

The access profile configuration is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show config access access-
profile
{
  "rtbrick-config:access-profile": [
    {
      "profile-name": "ipoe",
      "protocol": {
        "dhcp": {
          "enable": "true",
          "lease-time": 60
        },
        "dhcpv6": {
          "enable": "true",
          "lifetime": 60
```

```
        }
      },
      "address-family": {
        "ipv4": {
          "enable": "true",
          "pool-name": "poolv4",
          "instance": "default",
          "dad-enable": "true"
        },
        "ipv6": {
          "enable": "true",
          "pool-name": "poolv6",
          "prefix-delegation-pool-name": "poolv6pd",
          "instance": "default",
          "dad-enable": "true"
        }
      }
    }
  ]
}
supervisor@rtbrick>multiservice-edge.rtbrick.net: op>
```

2. Configure the Authentication and Accounting (AAA) profile for `aaa-profile`.

```
set access aaa-profile aaa-profile
set access aaa-profile aaa-profile session-timeout 0
set access aaa-profile aaa-profile idle-timeout 0
set access aaa-profile aaa-profile aaa-radius-profile radius-profile
set access aaa-profile aaa-profile authentication order RADIUS
set access aaa-profile aaa-profile accounting order RADIUS
set access aaa-profile aaa-profile accounting interim-interval 86400
set access aaa-profile aaa-profile accounting ingress accounting-source POLICER
set access aaa-profile aaa-profile accounting egress accounting-source CLASS
commit
```

The access AAA configuration is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show config access aaa-profile
{
  "rtbrick-config:aaa-profile": [
    {
      "profile-name": "aaa-profile",
      "session-timeout": 0,
      "idle-timeout": 0,
      "aaa-radius-profile": "radius-profile",
      "authentication": {
        "order": "RADIUS"
      },
      "accounting": {
```

```
        "order": "RADIUS",
        "interim-interval": 86400,
        "ingress": {
          "accounting-source": "POLICER"        },
        "egress": {
          "accounting-source": "CLASS"
        }
      }
    }
  ]
}
```

3. Configure the access interface. Double-tagged interface is configured in this case as the LAG interfaces (lag-1 for Multiservice Edge1 and lag-11 for Multiservice Edge2). The interface configuration assigns the access type, access profile, AAA profile, and further optional attributes like service-profile to the specified access interface.

Access interface (lag-1) commands for Multiservice Edge1.

```
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 service-profile-name
qos_service
set access interface double-tagged lag-1 1001 1100 1001 1100 aaa-profile-name aaa-
profile
set access interface double-tagged lag-1 1001 1100 1001 1100 gateway-ifl lo-0/0/1/0
set access interface double-tagged lag-1 1001 1100 1001 1100 redundancy-session-id 100
commit
```

The double-tagged lag-1 interface configuration for Multiservice Edge1 is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show config access interface
{
  "rtbrick-config:interface": {
    "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",
        "service-profile-name": "qos_service",
        "aaa-profile-name": "aaa-profile",
        "gateway-ifl": "lo-0/0/1/0"
        "redundancy-session-id": 100
      }
```

```
    ]
  }
}
```

Access interface (`lag-11`) commands for Multiservice Edge2.

```
set access interface double-tagged lag-11 1001 1100 1001 1100
set access interface double-tagged lag-11 1001 1100 1001 1100 access-type IPoE
set access interface double-tagged lag-11 1001 1100 1001 1100 access-profile-name ipoe
set access interface double-tagged lag-11 1001 1100 1001 1100 service-profile-name
qos_service
set access interface double-tagged lag-11 1001 1100 1001 1100 aaa-profile-name aaa-
profile
set access interface double-tagged lag-11 1001 1100 1001 1100 gateway-ifl lo-0/0/1/0
set access interface double-tagged lag-11 1001 1100 1001 1100 redundancy-session-id
200
commit
```

The double-tagged lag-11 interface configuration for Multiservice Edge.2 is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show config access interface
{
  "rtbrick-config:interface": {
    "double-tagged": [
      {
        "interface-name": "lag-11",
        "outer-vlan-min": 1001,
        "outer-vlan-max": 1100,
        "inner-vlan-min": 1001,
        "inner-vlan-max": 1100,
        "access-type": "IPoE",
        "access-profile-name": "ipoe",
        "service-profile-name": "qos_service",
        "aaa-profile-name": "aaa-profile",
        "gateway-ifl": "lo-0/0/1/0"
        "redundancy-session-id": 200
      }
    ]
  }
}
```

4. In this solution, we configure AAA authentication and accounting with RADIUS. To use RADIUS authentication and accounting both the RADIUS profile and RADIUS server configurations (see below) must be configured.

a. Configure RADIUS profile `radius-profile`.

```
set access radius-profile radius-profile
set access radius-profile radius-profile nas-ip-address 192.0.2.74
set access radius-profile radius-profile nas-port-format DEFAULT
set access radius-profile radius-profile nas-port-type Ethernet
set access radius-profile radius-profile authentication radius-server-profile-name
radius
set access radius-profile radius-profile accounting radius-server-profile-name radius
commit
```

The RADIUS profile configuration is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: op> show config access radius-
profile radius-profile
{
  "rtbrick-config:radius-profile": [
    {
      "profile-name": "radius-profile",
      "nas-ip-address": "192.0.2.74",
      "nas-port-format": "DEFAULT",
      "nas-port-type": "Ethernet",
      "authentication": {
        "radius-server-profile-name": [
          "radius"
          ]
      },
      "accounting": {
        "radius-server-profile-name": [
          "radius"
          ]
      }
    }
  ]
}
supervisor@rtbrick>multiservice-edge.rtbrick.net: op>
```

b. Configure the RADIUS server `radius`.

```
set access radius-server radius
set access radius-server radius address 192.0.2.34
set access radius-server radius source-address 192.0.2.74
set access radius-server radius secret-plain-text testing123
set access radius-server radius routing-instance default
set access radius-server radius rate 300
set access radius-server radius authentication enable true
set access radius-server radius authentication retry 3
set access radius-server radius authentication timeout 5
set access radius-server radius accounting enable true
set access radius-server radius accounting timeout 30
```

```
set access radius-server radius coa enable true
commit
```



The attribute` secret-plain-text` will be converted to `secret-encrypted-text` in the show command output and value will be hashed.

The RADIUS server configuration is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config access radius-
server radius
{
  "rtbrick-config:radius-server": [
    {
      "server-name": "radius",
      "address": "192.0.2.34",
      "source-address": "192.0.2.74",
      "secret-encrypted-text": "$2b2feb12f730107454b1be6a0f8242b0f",
      "routing-instance": "default",
      "rate": 300,
      "authentication": {
        "enable": "true",
        "retry": 3,
        "timeout": 5
      },
      "accounting": {
        "enable": "true",
        "timeout": 30
      },
      "coa": {
        "enable": "true"
      }
    }
  ]
}
```

5. Configure the IPv4 and IPv6 access pools.

```
set access pool poolv4
set access pool poolv4 ipv4-address low 203.0.113.1
set access pool poolv4 ipv4-address high 203.0.113.64
set access pool poolv6
set access pool poolv6 ipv6-prefix low 2001:db8:0:1::/64
set access pool poolv6 ipv6-prefix high 2001:db8:0:40::/64
set access pool poolv6pd
set access pool poolv6pd ipv6-prefix low 2001:db8:0:100::/56
set access pool poolv6pd ipv6-prefix high 2001:db8:0:4000::/56
commit
```

The access pool configuration is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config access pool
{
  "rtbrick-config:pool": [
    {
      "pool-name": "poolv4",
      "ipv4-address": {
        "low": "203.0.113.1",
        "high": "203.0.113.64"
      }
    },
    {
      "pool-name": "poolv6",
      "ipv6-prefix": {
        "low": "2001:db8:0:1::/64",
        "high": "2001:db8:0:40::/64"
      }
    },
    {
      "pool-name": "poolv6pd",
      "ipv6-prefix": {
        "low": "2001:db8:0:100::/56",
        "high": "2001:db8:0:4000::/56"
      }
    }
  ]
}
```

4.4.2. IPoE Quality of Service (QoS) Configuration



The QoS model explained in this document uses a complex HQoS model with the intent to showcase the complete range of QoS features available in RBFS. However, it may not be needed or desirable for all deployments. In such a case it should be possible to conceive of a simple QoS model as required by simplifying the provided QoS model.

Following are the steps involved in configuring and verifying IPoE QoS:

1. Configuring service profile to enable QoS on IPoE subscriber locally
2. Configuring downstream QoS
3. Configuring upstream QoS
4. Configuring QoS remarking
5. Configuring IPoE subscriber accounting for upstream and downstream traffic
6. Configuring IPoE subscribers QoS on BNG Blaster
7. Validating IPoE QoS on BNG Blaster

The figure below shows how QoS is configured for ingress and egress traffic.

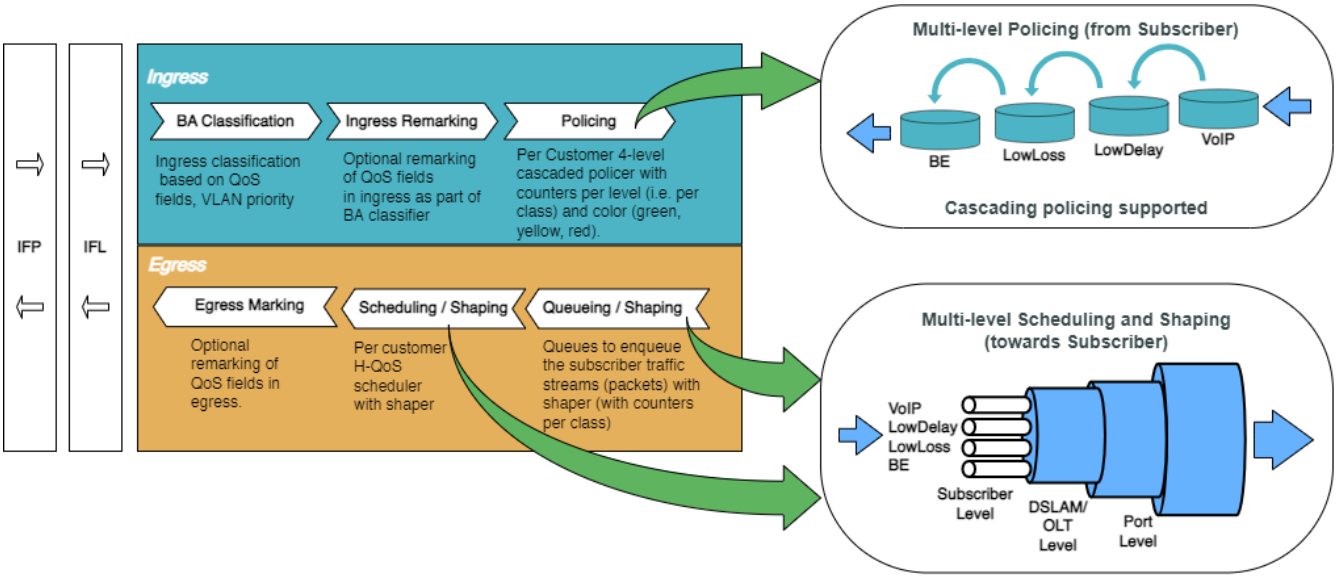


Figure 16. Hierarchical Quality of Service primitives

For detailed information about the QoS configuration options, see the HQoS Configuration Guide.

Configure Service Profile

Service profile configuration in subscriber management allows to assign QoS configurations to a subscriber.

1. Configure the service profile to enable QoS. The service profile defined to enable Quality of Service with profile name is `residential`.

```
set access service-profile qos_service qos profile residential
commit
```

The configuration of the service profile named `residential` is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config access service-
profile qos_service
{
  "rtbrick-config:service-profile": [
    {
      "profile-name": "qos_service",
      "qos": {
        "profile": "residential"
      }
    }
  ]
}
```

2. Enable QoS on IPoE subscriber access interface (`ifp-0/1/30`) to enable QoS for IPoE subscriber.

```
set access interface double-tagged ifp-0/1/30 1001 1100 1001 1100 service-profile-name
qos_service
commit
```

Below is the double-tagged access interface on which the service profile `qos_service` is configured.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config access interface
double-tagged ifp-0/1/30

{
  "rtbrick-config:double-tagged": [
    {
      "interface-name": "ifp-0/1/30",
      "outer-vlan-min": 1001,
      "outer-vlan-max": 1100,
      "inner-vlan-min": 1001,
      "inner-vlan-max": 1100,
      "access-type": "IPoE",
      "access-profile-name": "ipoe",
      "service-profile-name": "qos_service",
      "aaa-profile-name": "aaa-profile",
      "gateway-ifl": "lo-0/0/1/0",
      "redundancy-session-id": 100
    }
  ]
}
```

3. Configure QoS profile to enable on IPoE subscriber.

```
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 QoS Profile with all the primitives needed to enable traffic profiles on IPoE Subscribers is as follows:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config forwarding-options
class-of-service profile residential
{
  "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"
    }
  ]
}
```

Configure Downstream QoS

Downstream Quality of Service (QoS) is used to prioritize network traffic from the Internet to subscribers.

1. Enable global classification for downstream traffic.

```
set forwarding-options class-of-service global multifield-classifier-name global_mfc
commit
```

Below is the multi-field-classifier (MFC) based classifier for global enabling of downstream traffic classification.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config forwarding-options
class-of-service global multifield-classifier-name
{
  "rtbrick-config:multifield-classifier-name": "global_mfc"
}
```

2. Configure the MFC-based classifier with qualifiers and actions.

```
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
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1003
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1003 match ipv4-tos 192
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1003 match source-ipv4-prefix 132.1.1.3/32
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1003 action forward-class class-2
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1004
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1004 match ipv4-tos 224
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1004 match source-ipv4-prefix 132.1.1.3/32
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
ordinal 1004 action forward-class class-3
commit
```

The configuration of the QoS MFC-based Classifier for classification of downstream traffic from the core towards IPoE Subscriber is shown below.

```
supervisor@rtbrick>multiservice-edge.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"
            }
          },
          {
            "ordinal-value": 1003,
            "match": {
              "ipv4-tos": 192,
              "source-ipv4-prefix": "132.1.1.3/32"
            },
            "action": {
              "forward-class": "class-2"
            }
          },
          {
            "ordinal-value": 1004,
            "match": {
              "ipv4-tos": 224,
              "source-ipv4-prefix": "132.1.1.3/32"
            },
            "action": {
              "forward-class": "class-3"
            }
          }
        ]
      }
    ]
  }
```

3. Enqueue classified traffic to different queues using class-to-queue mapping.

```
set forwarding-options class-of-service queue-group subs-4queues queue-numbers 4
set forwarding-options class-of-service class-queue-map subs-4queues
set forwarding-options class-of-service class-queue-map subs-4queues class class-0
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
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
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
set forwarding-options class-of-service class-queue-map subs-4queues class class-3
queue-name VO_SUBS
commit
```

Below is the QoS class-queue mapping configuration:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config forwarding-options
class-of-service class-queue-map subs-4queues class
{
    "rtbrick-config: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"
      }
    ]
  }
```

4. Configure the queues needed for enqueuing and dequeuing traffic streams.

```
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 LD_SUBS
set forwarding-options class-of-service queue LD_SUBS queue-size 625000
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 VO_SUBS
set forwarding-options class-of-service queue VO_SUBS queue-size 156250
set forwarding-options class-of-service queue VO_SUBS shaper-name shaper_VO
commit
```

The queue Configuration is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config forwarding-options
class-of-service queue
{
  "rtbrick-config: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",
    }
  ]
}
```

5. Configure the scheduler needed by Subscriber/Session scheduler-map and OLT scheduler-map.

```
set forwarding-options class-of-service scheduler pon0
set forwarding-options class-of-service scheduler pon0 type fair_queueing
set forwarding-options class-of-service scheduler pon0 shaper-name gpon-shaper
set forwarding-options class-of-service scheduler subs-4queues
set forwarding-options class-of-service scheduler subs-4queues shaper-name
shaper_session
set forwarding-options class-of-service scheduler subs-4queues type strict_priority
set forwarding-options class-of-service scheduler subs-4queues composite false
commit
```

The configuration of the scheduler-map and OLT scheduler-map is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config  forwarding-options
class-of-service scheduler
{
  "rtbrick-config:scheduler": [
    {
      "scheduler-name": "pon0",
      "shaper-name": "gpon-shaper",
      "type": "fair_queueing"
    },
    {
      "scheduler-name": "subs-4queues",
      "shaper-name": "shaper_session",
      "type": "strict_priority",
      "composite": "false"
    }
  ]
}
```

6. Configure the session/subscriber scheduler mapping for dequeuing traffic based on scheduler type for each queue:

```
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 QoS Subscriber/Session Scheduler-Map configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config forwarding-options
class-of-service scheduler-map subs-4queues-residential
```

```
{
  "rtbrick-config:scheduler-map": [
    {
      "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"
        }
      ]
    }
  ]
}
```

7. Configure the OLT scheduler-mapping for each PON to be scheduled according to the scheduler type.

```
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
commit
```

The OLT Scheduler-Map configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config forwarding-options
class-of-service scheduler-map schedmap-olt
{
  "rtbrick-config:scheduler-map": [
    {
      "scheduler-map-name": "schedmap-olt",
      "scheduler-name": [
        {
          "name": "pon0",
          "port-connection": "scheduler_to_port"
        }
      ]
    }
  ]
}
```

8. Configure downstream traffic shaping for both session schedulers and queues.



Queue Shaping is only on VO_SUBS Queue session.

```
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
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 gpon-shaper
set forwarding-options class-of-service shaper gpon-shaper shaping-rate-high 2488000
set forwarding-options class-of-service shaper gpon-shaper shaping-rate-low 32000
commit
```

The shaping Configuration is shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: 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
    },
    {
      "shaper-name": "gpon-shaper",
      "shaping-rate-high": 2488000,
      "shaping-rate-low": 32000
    }
  ]
}
```

Configure Upstream QoS

1. Configure the BA Classifier for the classification of multiple traffic streams targeted at IPoE subscribers:

```
set forwarding-options class-of-service classifier subs-pbit-class
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 1
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 1 class class-0
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 1 remark-codepoint 7
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 2
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 2 class class-1
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 2 remark-codepoint 7
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 3
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 3 class class-2
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 3 remark-codepoint 7
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 4
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 4 class class-3
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 4 remark-codepoint 7
commit
```

The configuration of the QoS BA-based Classifier for classification of upstream traffic towards IPoE

Subscriber is shown below.

```
 supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config forwarding-options
class-of-service classifier subs-pbit-class
{
  "rtbrick-config: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
            }
          ]
        }
      ]
    }
  ]
}
```

2. Configure multi-level policer to police 4-Level traffic.

```
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 multi-level policer configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: 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"
    }
  ]
}
```

3. Map the classified traffic streams to different policer levels using class-to-policer mapping:

```
set forwarding-options class-of-service class-policer-map policer-map-residential
set forwarding-options class-of-service class-policer-map policer-map-residential
class class-0
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
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
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
set forwarding-options class-of-service class-policer-map policer-map-residential
class class-3 policer-level level-4
commit
```

The class-policer-map configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config forwarding-options
class-of-service class-policer-map policer-map-residential class
{
  "rtbrick-config: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"
    }
  ]
}
```

Configure QoS Remarking

1. Remark downstream traffic egressing from subscriber interface (egress remarking).

```
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 1
set forwarding-options class-of-service remark-map subs-remarking-residential remark-
type ieee-802.1 match-codepoint 1 color all
set forwarding-options class-of-service remark-map subs-remarking-residential remark-
type ieee-802.1 match-codepoint 1 color all remark-codepoint 6
set forwarding-options class-of-service remark-map subs-remarking-residential remark-
type ieee-802.1 match-codepoint 2
set forwarding-options class-of-service remark-map subs-remarking-residential remark-
type ieee-802.1 match-codepoint 2 color all
set forwarding-options class-of-service remark-map subs-remarking-residential remark-
type ieee-802.1 match-codepoint 2 color all remark-codepoint 6
set forwarding-options class-of-service remark-map subs-remarking-residential remark-
type ieee-802.1 match-codepoint 3
set forwarding-options class-of-service remark-map subs-remarking-residential remark-
type ieee-802.1 match-codepoint 3 color all
set forwarding-options class-of-service remark-map subs-remarking-residential remark-
type ieee-802.1 match-codepoint 3 color all remark-codepoint 6
set forwarding-options class-of-service remark-map subs-remarking-residential remark-
type ieee-802.1 match-codepoint 4
set forwarding-options class-of-service remark-map subs-remarking-residential remark-
type ieee-802.1 match-codepoint 4 color all
set forwarding-options class-of-service remark-map subs-remarking-residential remark-
type ieee-802.1 match-codepoint 4 color all remark-codepoint 6
commit
```

The remarking configuration is shown below:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config forwarding-options
class-of-service remark-map subs-remarking-residential
{
  "rtbrick-config:remark-map": [
    {
      "remark-map-name": "subs-remarking-residential",
      "remark-type": [
        {
          "remark-type": "ieee-802.1",
          "match-codepoint": [
            {
              "match-codepoint": 1,
              "color": [
                {
                  "color": "all",
                  "remark-codepoint": 6
                }
              ]
            },
            {
              "match-codepoint": 2,
              "color": [
                {
                  "color": "all",
                  "remark-codepoint": 6
                }
              ]
            },
            {
              "match-codepoint": 3,
              "color": [
                {
                  "color": "all",
                  "remark-codepoint": 6
                }
              ]
            },
            {
              "match-codepoint": 4,
              "color": [
                {
                  "color": "all",
                  "remark-codepoint": 6
                }
              ]
            }
          ]
        }
      ]
    }
```

```
  ]
}
```

3. Remark upstream traffic ingressing to a subscriber's interface [ingress remarking]



In the upstream traffic classifier configuration shown below, remarking of all traffic streams with code point '7' is done.

```
set forwarding-options class-of-service classifier subs-pbit-class
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 1
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 1 class class-0
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 1 remark-codepoint 7
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 2
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 2 class class-1
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 2 remark-codepoint 7
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 3
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 3 class class-2
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 3 remark-codepoint 7
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 4
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 4 class class-3
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-
802.1 codepoint 4 remark-codepoint 7
commit
```

Below is the remarking configuration:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config forwarding-options
class-of-service classifier subs-pbit-class
{
  "rtbrick-config: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
            }
          ]
        }
      ]
    }
  ]
}
```

4.4.3. Configure Mirroring for Downstream Traffic Remark Validation

Configure Mirroring for Downstream Traffic

```
set forwarding-options mirror m1 source interface ifp-0/1/30
set forwarding-options mirror m1 source direction egress
set forwarding-options mirror m1 destination interface cpu-0/0/200
commit
```

Validate the downstream traffic remarking



This validation requires mirroring the subscriber access interface on the Multiservice Edge device.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config forwarding-options
mirror
{
  "rtbrick-config:mirror": [
    {
      "name": "m1",
```

```
      "destination": {
        "interface": "cpu-0/0/200"
      },
      "source": {
        "direction": "egress",
        "interface": "ifp-0/1/30"
      }
    }
  ]
}
```

The capture mirroring can be performed as shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> capture mirror

2023-04-03T13:58:59.651227+0000 e8:c5:7a:8f:76:f1 > 02:00:00:00:00:01, ethertype
802.1Q (0x8100), length 1022: vlan 1001, p 6, ethertype 802.1Q, vlan 1001, p 6,
ethertype IPv4, (tos 0xe0, ttl 63, id 0, offset 0, flags [DF], proto UDP (17), length
1000)
    192.0.2.2.65056 > 203.0.113.1.65056: UDP, length 972

2023-04-03T13:58:59.651306+0000 e8:c5:7a:8f:76:f1 > 02:00:00:00:00:01, ethertype
802.1Q (0x8100), length 1022: vlan 1001, p 6, ethertype 802.1Q, vlan 1001, p 6,
ethertype IPv4, (tos 0xe0, ttl 63, id 0, offset 0, flags [DF], proto UDP (17), length
1000)
    192.0.2.2.65056 > 203.0.113.1.65056: UDP, length 972
```

Configure Mirroring for Upstream Traffic Remark Validation

```
set forwarding-options mirror m1 source interface ifp-0/1/25
set forwarding-options mirror m1 source direction ingress
set forwarding-options mirror m1 destination interface cpu-0/0/200
commit
```

Validating the upstream traffic remarking



Mirror the core facing port on the Multiservice Edge device as shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show config forwarding-options
mirror m1
{
  "rtbrick-config:mirror": [
    {
      "name": "m1",
      "destination": {
        "interface": "cpu-0/0/200"
      },
```

```
      "source": {
        "direction": "ingress",
        "interface": "ifp-0/1/25"
      }
    }
  ]
}
```

The capture mirroring can be performed as shown below. It confirms all four traffic streams noted with codepoint=7.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> capture mirror

2023-04-03T13:53:31.400011+0000 02:00:00:00:00:01 > e8:c5:7a:8f:76:f1, ethertype
802.1Q (0x8100), length 1022: vlan 1001, p 2, ethertype 802.1Q, vlan 1001, p 2,
ethertype IPv4, (tos 0xa0, ttl 64, id 0, offset 0, flags [DF], proto UDP (17), length
1000)
    203.0.113.1.65056 > 192.0.2.2.65056: UDP, length 972

2023-04-03T13:53:31.400167+0000 02:00:00:00:00:01 > e8:c5:7a:8f:76:f1, ethertype
802.1Q (0x8100), length 1022: vlan 1001, p 3, ethertype 802.1Q, vlan 1001, p 3,
ethertype IPv4, (tos 0xc0, ttl 64, id 0, offset 0, flags [DF], proto UDP (17), length
1000)
    203.0.113.1.65056 > 192.0.2.2.65056: UDP, length 972
```

4.4.4. Configure FreeRADIUS Server

Installation of FreeRADIUS

FreeRADIUS server can be installed on any Linux OS distribution. For information about installing FreeRADIUS, see Installing the FreeRADIUS Server.

Remove the Unsupported Files

It is necessary to remove `echo`, `ntlm_auth`, `eap`, `echo`, and `mschap` files once FreeRadius has been installed, since they are not required by this reference design. To remove these files, enter the following commands:

```
rm -rf /etc/freeradius/3.0/mods-enabled/echo
rm -rf /etc/freeradius/3.0/mods-enabled/ntlm_auth
rm -rf /etc/freeradius/3.0/mods-enabled/eap
rm -rf /etc/freeradius/3.0/mods-enabled/echo
rm -rf /etc/freeradius/3.0/mods-enabled/mschap
```

FreeRADIUS User and Group Settings

Using the following commands, one can set the user or group as root.

```
sed -i 's/User=freerad/User=root/g' /lib/systemd/system/freeradius.service
sed -i 's/Group=freerad/Group=root/g' /lib/systemd/system/freeradius.service
```

Run the command for reloading 'systemd' after user or group settings:

```
systemctl daemon-reload
```

Configure the FreeRADIUS Files

It is necessary to configure the FreeRADIUS files once FreeRadius has been installed. The configuration files can be found under `/etc/freeradius/3.0/`.

authorize

Using the following command, one can view the `authorize` file in its default location.

```
~:/etc/freeradius/3.0 # cat mods-config/files/authorize
```

Replace the content of the `authorize` file with the following:

```
$INCLUDE /etc/freeradius/3.0/ipoe_users_file
```



This file can also be downloaded from the appendix (Appendix C: RADIUS Server Configuration).

ipoe_users_file

The IPoE Users file (`ipoe_users_file`) mentioned in the above section includes subscriber profile parameters as shown below.

```
"02:00:00:00:00:01@ipoe" Cleartext-Password := "ipoe"
    Service-Type = Framed-User,
    Class = IPOE,
    Framed-IP-Address = 203.0.113.1,
    Framed-IP-Netmask = 255.255.255.255,
    RtBrick-DNS-Primary = 203.0.113.200,
    RtBrick-DNS-Secondary  = 203.0.113.201,
    Framed-IPv6-Prefix = 2001:db8:0:1::1/128,
    Delegated-IPv6-Prefix = 2001:db8:0:3::/64,
    RtBrick-DNS-Primary-IPv6  = 2001:db8:0:20::1,
    RtBrick-DNS-Secondary-IPv6 = 2001:db8:0:20::2,
    Session-Timeout = 0,
    Idle-Timeout = 0,
```

```
    Reply-Message = "FOOBAR Internet",
    Acct-Interim-Interval = 120,
    RtBrick-QoS-Profile = "residential"

DEFAULT User-Name =~ '^[0-9a-f\:]+@ipoe$', Cleartext-Password := 'ipoe'
    Service-Type = Framed-User,
    Class = "IPOE",
    Acct-Interim-Interval = 900
```

The `ipoe_users_file` can be created with the above content in the `/etc/freeradius/3.0/` path. Alternatively, this file can be downloaded from the appendix section of this guide and placed at `/etc/freeradius/3.0/`.

clients.conf

Clients.conf file shall be configured with the expected RADIUS client IP address and secret.

```
~:/etc/freeradius/3.0 # cat clients.conf
client rtbrick {
        ipaddr          = 192.0.2.74
        secret          = testing123
        shortname       = rtbrick
        nas_type        = other
        require_message_authenticator = no
}
```

The `clients.conf` file (/etc/freeradius/3.0/clients.conf) used for this reference design can be downloaded from the appendix section of this guide.

radiusd.conf

The `radiusd.conf` file should be configured with the expected RADIUS authentication and accounting parameters.

```
prefix = /usr
exec_prefix = /usr
sysconfdir = /etc
localstatedir = /var
sbindir = ${exec_prefix}/sbin
logdir = /var/log/freeradius
raddbdir = /etc/freeradius
radacctdir = ${logdir}/radacct
name = freeradius
confdir = ${raddbdir}
modconfdir = ${confdir}/mods-config
run_dir = ${localstatedir}/run/${name}
db_dir = ${raddbdir}
```

```
libdir = /usr/lib/freeradius
pidfile = ${run_dir}/${name}.pid
correct_escapes = true
max_request_time = 5
cleanup_delay = 0
max_requests = 16384
hostname_lookups = no
log {
    destination = files
    file = ${logdir}/radius.log
    stripped_names = no
    auth = yes
}
checkrad = ${sbindir}/checkrad
security {
    # user = radius
    # group = radius
    allow_core_dumps = no
    max_attributes = 200
    reject_delay = 0
    status_server = no
@openssl_version_check_config@
}
proxy_requests  = no
$INCLUDE clients.conf
thread pool {
    start_servers = 32
    max_servers = 128
    min_spare_servers = 8
    max_spare_servers = 16
    max_queue_size = 16384
    max_requests_per_server = 0
    auto_limit_acct = no
}
modules {
    $INCLUDE mods-enabled/
}
instantiate {
    files
    linelog
}
server default {
    listen {
        type = auth
        ipaddr = *
        port = 1812
    }
    listen {
        type = acct
        ipaddr = *
        port = 1813
```

```
    }
    authorize {
        update request {
            FreeRADIUS-Client-Shortname = "%{&request:Client-Shortname}"
        }
        if (&request:Client-Shortname == "rtbrick-server") {
            rtbrick-server-log
        }
        files
        pap
    }
    authenticate {
        pap
    }
    post-auth {
        if (&request:Client-Shortname == "rtbrick-server") {
            rtbrick-server-log
        }
        Post-Auth-Type REJECT {
            update reply {
                Reply-Message := "Login Failed. Please check your username and
password."
            }
            attr_filter.access_reject
        }
    }
    preacct {
        ok
    }
    accounting {
        if (&request:Client-Shortname == "rtbrick-server") {
            rtbrick-server-log
        }
    ok
    }
    session {
    }
}
```

Radiusd.conf should be configured to use UDP ports 1812 and 1813 for authentication and accounting, respectively. Additionally, `rtbrick-server-log` should be added to the parameters for authorise, authenticate, and accounting.

The `radiusd.conf` file (/etc/freeradius/3.0/radiusd.conf) used for this reference design can be downloaded from the appendix section (Appendix C) of this guide.

detail

The `detail` file shall be configured for the RADIUS accounting logs.

```
/etc/freeradius/3.0 # cat mods-enabled/detail
    permissions = 0666detail rtbrick-server-log {
    filename = ${radacctdir}/rtbrick-server-detail.log

    header = "%t;%{NAS-IP-Address};%I;%{Packet-Src-Port}"
    log_packet_header = no
}
```

Ensure that `rtbrick-server-log` is specified in the `detail` file.

The `detail` file (/etc/freeradius/3.0/mods-enabled/detail) used for this reference design can be downloaded from the appendix section (Appendix C) of this guide.

dictionary.rtbrick

Add the RtBrick RADIUS dictionary (dictionary.rtbrick) to `/usr/share/freeradius/dictionary.rtbrick` and include it in` /usr/share/freeradius/dictionary`.

The `dictionary.rtbrick` contains the RBFS attributes in FreeRADIUS format.

- Click here to download the `radius_config.zip`, which contains the `dictionary.rtbrick` file.

Stopping and Starting the FreeRADIUS Server for any Changes

For any changes, stop and restart the FreeRADIUS server.

To stop the server, enter the following command:

```
sudo service freeradius stop
```

To start the server, enter the following command:

```
sudo service freeradius start
```

The FreeRadius server is now ready to provide AAA (Authentication, Accounting & Authorization) services to logging in subscribers.

Displaying the RADIUS Service Status

Run the following command to determine whether RADIUS service is active:

```
sudo service freeradius status
```

4.4.5. Validating IPoE Subscriber Bring-Up

Using traffic streams on both upstream and downstream directions with traffic packets and bytes statistics, IPoE Subscriber sessions can be "ESTABLISHED".

The validation can be performed in two steps:

1. Establishing the IPoE subscriber
2. Pinging the subscriber IPv4/IPv6 address

BNG Blaster - IPoE Subscribers with Traffic Streams

Using BNG Blaster, which emulates IPoE clients, session traffic("session-traffic") and streams traffic with different code-points("streams"), one can test IPoE subscriber management feature.

The data traffic can be defined in Blaster by configuring session/streams traffic.

- Session Traffic in Blaster can be enabled by specifying `"autostart": true`. Once the session is established, traffic starts automatically. Eventually if you want to stop or start the traffic (session or streams), press F7/F8.
- Streams traffic in Blaster can be enabled by specifying `stream-group-id` at both streams (`"streams":`) and access interface levels (`interfaces:access`).

For information about using BNG Blaster, see Starting BNG Blaster.

Validating the IPoE Session on BNG Blaster

To validate the IPoE Session on BNG Blaster, switch to the Service Node. On the BNG Blaster terminal, the count of the "Established" session will be 1. Also, the logging of the same terminal will display "ALL SESSIONS ESTABLISHED" and "ALL SESSION TRAFFIC FLOWS VERIFIED" as highlighted in the below image.

In the image below, one can see the details of the established Single Subscriber session.

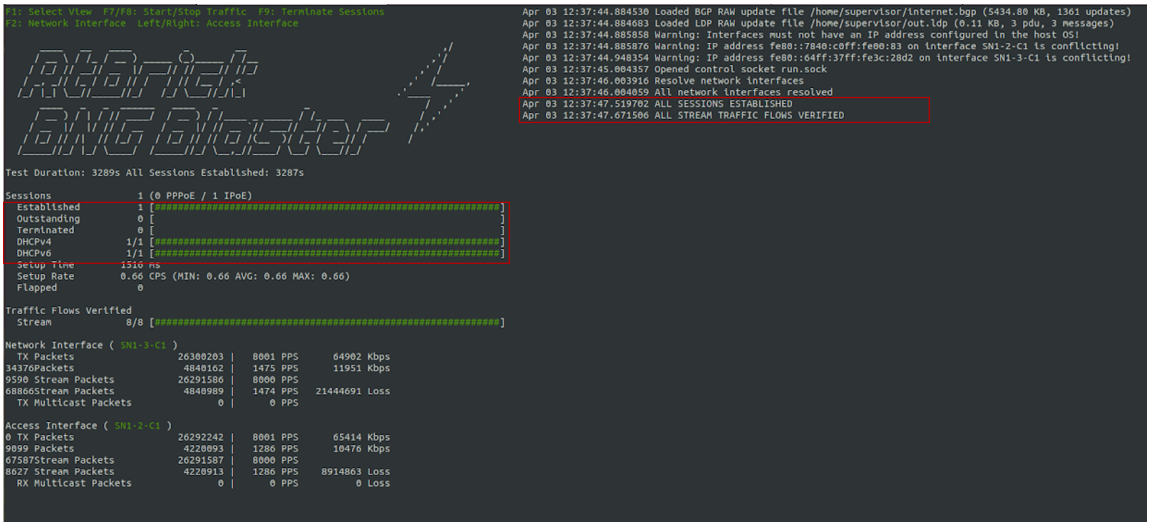


Figure 17. BNG Blaster Terminal View

Visit the following URL for more information on BNG Blaster:

https://rtbrick.github.io/bngblaster/

Viewing the Subscribers and the Subscriber Details on RBFS

Enter the following command to view the list of subscribers.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show subscriber
Subscriber-Id          Interface        VLAN      Type   State
216454257090494470     ifp-0/1/30       1001:1001 IPoE   ESTABLISHED
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg>
```

Enter the following command to view the details of the subscriber with ID 216454257090494470.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show subscriber
216454257090494470 detail
Subscriber-Id: 216454257090494470
    Type: IPoE
    State: ESTABLISHED
    Created: Mon Apr 03 13:51:17 GMT +0000 2023
    Interface: ifp-0/1/30
    Outer VLAN: 1001
    Inner VLAN: 1001
    Client MAC: 02:00:00:00:00:01
    Server MAC: e8:c5:7a:8f:76:f1
    IFL: ipoe-0/1/30/216454257090494470
    Username: 02:00:00:00:00:01@ipoe
    Access-Profile: ipoe
    AAA-Profile: aaa-profile
    Service-Profile: qos_service
    Reply-Message: FOOBAR Internet
    Session-Timeout: 0 (disabled)
    Idle-Timeout: 0 (disabled)
    MTU: 1500 Profile: N/A
    IPv4:
        Instance: default
        Address: 203.0.113.1/255.255.255.255
        Address Active: True
        Primary DNS: 203.0.113.200
        Secondary DNS: 203.0.113.201
    IPv6:
        Instance: default
        RA Prefix: 2001:db8:0:1::1/128
        RA Prefix Active: True
        Delegated Prefix (DHCPv6): 2001:db8:0:3::/64
        Delegated Prefix Active: True
        Primary DNS: 2001:db8:0:20::1
        Secondary DNS: 2001:db8:0:20::2
    Accounting:
        Session-Id: 216454257090494470:1680529877
```

```
        Start-Time: 2023-04-03T13:51:18.543868+0000
        Interims Interval: 120 seconds
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg>
```

Pinging the Subscriber (source: IPOE) from Multiservice Edge

Before pinging a subscriber, use the `show route <···>` command to display the subscriber IPs at the Multiservice Edge.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show route ipv4 unicast source
ipoe
Instance: default, AFI: ipv4, SAFI: unicast
Prefix/Label                            Source            Pref    Next Hop
Interface
203.0.113.1/32                          ipoe              7       -
ipoe-0/1/30/216454257090494470
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg>
```

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show route ipv6 unicast source
ipoe
Instance: default, AFI: ipv6, SAFI: unicast
Prefix/Label                            Source            Pref    Next Hop
Interface
2001:db8:0:1::1/128                     ipoe              7       -
ipoe-0/1/30/216454257090494470
2001:db8:0:3::/64                       ipoe              7
fe80::ffff:ffff:ff00:1                  ipoe-0/1/30/216454257090494470
```

From the above list, ping `203.0.113.1` as shown below.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> ping 203.0.113.1
68 bytes from 203.0.113.1: icmp_seq=1 ttl=64 time=1.3463 ms
68 bytes from 203.0.113.1: icmp_seq=2 ttl=64 time=10.5234 ms
68 bytes from 203.0.113.1: icmp_seq=3 ttl=64 time=9.8053 ms
68 bytes from 203.0.113.1: icmp_seq=4 ttl=64 time=11.0041 ms
68 bytes from 203.0.113.1: icmp_seq=5 ttl=64 time=10.9300 ms
Statistics: 5 sent, 5 received, 0% packet loss
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg>
```

Validating Traffic Streams

Traffic streams can be used to perform various forwarding verifications.

For upstream traffic capture, enter the following command:

```
capture interface ifp-0/1/30 direction in
```

For downstream traffic capture, enter the following command:

```
capture interface ifp-0/1/30 direction out
```

Here, `ifp-0/1/30` refers to the access interface.

Validating the IPoE QoS on BNG Blaster

To validate the IPoE QoS on BNG Blaster, switch to the Service Node. Navigate to the Streams and Session Traffic terminal by pressing F1 function key in your keyboard.

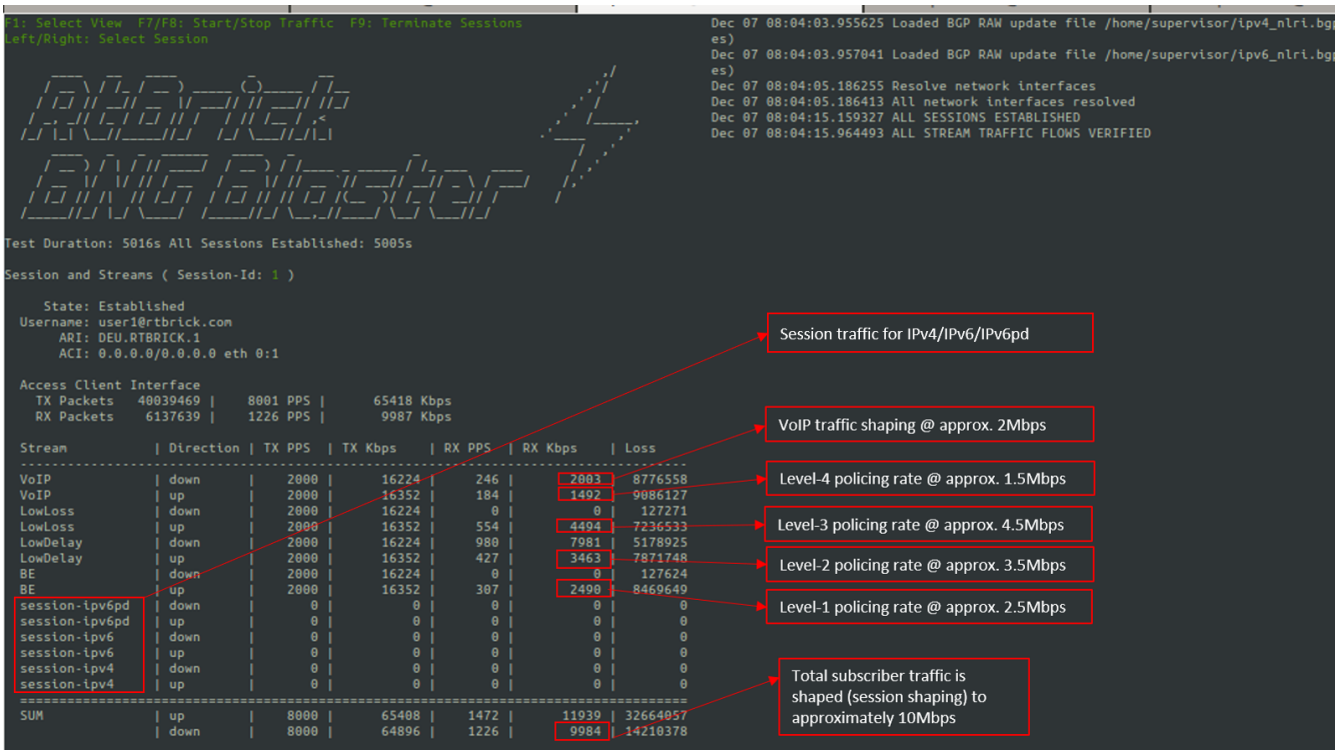


Figure 18. Reading output from BNG Blaster

As shown in the above image, the VoIP downstream traffic has been shaped (session shaping) to 2Mbps. Similarly, the total subscriber traffic has been shaped approximately to 10Mbps.

Following are the upstream traffic rates of different policer levels:

- Level-1 Rates ~=2.5Mbps
- Level-2 Rates ~=3.5Mbps
- Level-3 Rates ~=4.5Mbps
- Level-4 Rates ~=1.5Mbps

4.4.6. IPoE Subscriber Accounting for Upstream and Downstream Traffic

Run the "show subscriber" command to view the list of subscribers.

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show subscriber
Subscriber-Id          Interface        VLAN      Type   State
```

```
216454257090494470     ifp-0/1/30       1001:1001 IPoE   ESTABLISHED
ESTABLISHEDsupervisor@rtbrick>multiservice-edge.rtbrick.net: cfg>
```

Specify the Subscriber-ID to find the specific subscriber's accounting details:

```
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg> show subscriber
216454257090494470 accounting
Subscriber-Id: 216454257090494470
    IFL: ipoe-0/1/30/216454257090494470
    Start Timestamp: Mon Apr 03 13:51:18 GMT +0000 2023
    Idle Timestamp: Mon Apr 03 13:51:38 GMT +0000 2023
    Session-Timeout: 0 seconds
    Idle-Timeout: 0 seconds
    Session Statistics:
        Ingress: 28185 packets 27339450 bytes
        Egress: 23990 packets 23990000 bytes
    LIF Statistics:
        Ingress: 0 packets 0 bytes
        Egress: 0 packets 0 bytes
    Egress Class (Queue) Statistics:
        class-0: 4 packets 4000 bytes dropped: 0 packets 0 bytes
        class-1: 19191 packets 19191000 bytes dropped: 0 packets 0 bytes
        class-2: 3 packets 3000 bytes dropped: 0 packets 0 bytes
        class-3: 4792 packets 4792000 bytes dropped: 0 packets 0 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: 5819 packets 5644430 bytes dropped: 31270 packets 30331900 bytes
        Level 2: 8170 packets 7924900 bytes dropped: 29029 packets 28158130 bytes
        Level 3: 10540 packets 10223800 bytes dropped: 26638 packets 25838860 bytes
        Level 4: 3656 packets 3546320 bytes dropped: 33614 packets 32605580 bytes
supervisor@rtbrick>multiservice-edge.rtbrick.net: cfg>
```

4.5. RBFS Redundancy Configurations

4.5.1. Configure RBFS Redundancy

To deploy RBFS in redundancy mode, you must complete the following configuration on both Multiservice Edge nodes which are paired for redundancy.

- Redundancy Profile
- Redundancy Session
- Link Aggregation Group for the Redundancy Session
- Access for the Redundancy Session

The following steps provide the commands to configure RBFS Redundancy on both devices, which comprise the redundancy pair. For more detailed information about RBFS Redundancy Deployment, see RBFS Redundancy Solution Guide.

Redundancy Profile Configuration

Run the following commands to configure the redundancy profile on both devices (CBNG1 and CBNG2). Redundancy profile configuration is required to enable the devices to identify the peer in the network.

```
set redundancy profile rd_ipoe_olt1
set redundancy profile rd_ipoe_olt1 switchover-hold-timer 0
set redundancy profile rd_ipoe_olt1 peer ipv4 remote-address 192.0.2.200
set redundancy profile rd_ipoe_olt1 peer ipv4 update-source 192.0.2.100
set redundancy profile rd_ipoe_olt1 peer ipv4 instance default
set redundancy profile rd_ipoe_olt2
set redundancy profile rd_ipoe_olt2 switchover-hold-timer 0
set redundancy profile rd_ipoe_olt2 peer ipv4 remote-address 192.0.2.20
set redundancy profile rd_ipoe_olt2 peer ipv4 update-source 192.0.2.10
set redundancy profile rd_ipoe_olt2 peer ipv4 instance default
```

Redundancy Session Configuration

Run the following commands to configure the session for redundancy. Redundancy sessions can be uniquely identified by session IDs, which are used to store subscriber-specific data between active and standby devices.

```
set redundancy session 100
set redundancy session 100 priority 20
set redundancy session 100 profile rd_ipoe_olt1
set redundancy session 200
set redundancy session 200 priority 210
set redundancy session 200 profile rd_ipoe_olt2
```

Redundancy configuration from the node `CBNG1` is shown below:

```
supervisor@rtbrick>cbng1.rtbrick.net: cfg> show config redundancy
{
  "rtbrick-config:redundancy": {
    "profile": [
      {
        "name": "rd_ipoe_olt1",
        "switchover-hold-timer": 0,
        "peer": {
          "ipv4": {
            "remote-address": "192.0.2.200",
            "update-source": "192.0.2.100",
            "instance": "default"
```

```
          }
        }
      },
      {
        "name": "rd_ipoe_olt2",
        "switchover-hold-timer": 0,
        "peer": {
          "ipv4": {
            "remote-address": "192.0.2.20",
            "update-source": "192.0.2.10",
            "instance": "default"
          }
        }
      }
    ],
    "session": [
      {
        "session-id": 100,
        "priority": 20,
        "profile": "rd_ipoe_olt1"
      },
      {
        "session-id": 200,
        "priority": 210,
        "profile": "rd_ipoe_olt2"
      }
    ]
  }
}
```

The redundancy configuration from the node CBNG2 is shown below:

```
supervisor@rtbrick>cbng2.rtbrick.net: cfg> show config redundancy
{
  "rtbrick-config:redundancy": {
    "profile": [
      {
        "name": "rd_ipoe_olt1",
        "switchover-hold-timer": 0,
        "peer": {
          "ipv4": {
            "remote-address": "192.0.2.100",
            "update-source": "192.0.2.200",
            "instance": "default"
          }
        }
      },
      {
        "name": "rd_ipoe_olt2",
        "switchover-hold-timer": 0,
```

```
        "peer": {
          "ipv4": {
            "remote-address": "192.0.2.10",
            "update-source": "192.0.2.20",
            "instance": "default"
          }
        }
      }
    ],
    "session": [
      {
        "session-id": 100,
        "priority": 10,
        "profile": "rd_ipoe_olt1"
      },
      {
        "session-id": 200,
        "priority": 220,
        "profile": "rd_ipoe_olt2"
      }
    ]
  }
}
```



The session `priority` value determines the active/standby nodes for a session.

LAG Configuration for Redundancy

Run the following commands to configure LAG on both devices (`CBNG1` and `CBNG2`) for redundancy. This configuration is required to associate the redundancy session with LAG.

Run the following commands to configure LAG on `CBNG1` 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 3
set link-aggregation interface lag-1 redundancy-session-id 100
set link-aggregation interface lag-1 system-id a0:aa:aa:aa:aa:aa
set link-aggregation interface lag-1 member-interface ifp-0/1/26
set link-aggregation interface lag-1 member-interface ifp-0/1/27
set link-aggregation interface lag-1 member-interface ifp-0/1/28
set link-aggregation interface lag-11
set link-aggregation interface lag-11 mode lacp
set link-aggregation interface lag-11 minimum-link-count 3
set link-aggregation interface lag-11 redundancy-session-id 200
set link-aggregation interface lag-11 system-id a0:aa:aa:aa:aa:a0
set link-aggregation interface lag-11 member-interface ifp-0/1/31
set link-aggregation interface lag-11 member-interface ifp-0/1/33
set link-aggregation interface lag-11 member-interface ifp-0/1/35
```

```
set link-aggregation interface lag-2
set link-aggregation interface lag-2 mode lacp
set link-aggregation interface lag-2 minimum-link-count 1
set link-aggregation interface lag-2 member-interface ifp-0/1/14
set link-aggregation interface lag-2 member-interface ifp-0/1/15
set link-aggregation interface lag-3
set link-aggregation interface lag-3 mode lacp
set link-aggregation interface lag-3 minimum-link-count 1
set link-aggregation interface lag-3 member-interface ifp-0/1/23
set link-aggregation interface lag-3 member-interface ifp-0/1/24
```

LAG configuration for redundancy on CBNG1 is shown below:

```
supervisor@rtbrick>cbng1.rtbrick.net: cfg> show config link-aggregation
{
  "rtbrick-config:link-aggregation": {
    "interface": [
      {
        "interface-name": "lag-1",
        "mode": "lacp",
        "minimum-link-count": 3,
        "redundancy-session-id": 100,
        "system-id": "a0:aa:aa:aa:aa:aa",
        "member-interface": [
          {
            "member-interface-name": "ifp-0/1/26"
          },
          {
            "member-interface-name": "ifp-0/1/27"
          },
          {
            "member-interface-name": "ifp-0/1/28"
          }
        ]
      },
      {
        "interface-name": "lag-11",
        "mode": "lacp",
        "minimum-link-count": 3,
        "redundancy-session-id": 200,
        "system-id": "a0:aa:aa:aa:aa:a0",
        "member-interface": [
          {
            "member-interface-name": "ifp-0/1/31"
          },
          {
            "member-interface-name": "ifp-0/1/33"
          },
          {
            "member-interface-name": "ifp-0/1/35"
```

```
          }
        ]
      },
      {
        "interface-name": "lag-2",
        "mode": "lacp",
        "minimum-link-count": 1,
        "member-interface": [
          {
            "member-interface-name": "ifp-0/1/14"
          },
          {
            "member-interface-name": "ifp-0/1/15"
          }
        ]
      },
      {
        "interface-name": "lag-3",
        "mode": "lacp",
        "minimum-link-count": 1,
        "member-interface": [
          {
            "member-interface-name": "ifp-0/1/23"
          },
          {
            "member-interface-name": "ifp-0/1/24"
          }
        ]
      }
    ]
  }
}
```

Run the following commands to configure LAG on CBNG2 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 3
set link-aggregation interface lag-1 redundancy-session-id 100
set link-aggregation interface lag-1 system-id a0:aa:aa:aa:aa:aa
set link-aggregation interface lag-1 member-interface ifp-0/1/26
set link-aggregation interface lag-1 member-interface ifp-0/1/27
set link-aggregation interface lag-1 member-interface ifp-0/1/28
set link-aggregation interface lag-11
set link-aggregation interface lag-11 mode lacp
set link-aggregation interface lag-11 minimum-link-count 3
set link-aggregation interface lag-11 redundancy-session-id 200
set link-aggregation interface lag-11 system-id a0:aa:aa:aa:aa:a0
set link-aggregation interface lag-11 member-interface ifp-0/1/31
set link-aggregation interface lag-11 member-interface ifp-0/1/33
```

```
set link-aggregation interface lag-11 member-interface ifp-0/1/35
set link-aggregation interface lag-2
set link-aggregation interface lag-2 mode lacp
set link-aggregation interface lag-2 minimum-link-count 1
set link-aggregation interface lag-2 member-interface ifp-0/1/14
set link-aggregation interface lag-2 member-interface ifp-0/1/15
set link-aggregation interface lag-3
set link-aggregation interface lag-3 mode lacp
set link-aggregation interface lag-3 minimum-link-count 1
set link-aggregation interface lag-3 member-interface ifp-0/1/23
set link-aggregation interface lag-3 member-interface ifp-0/1/24
```

LAG configuration for redundancy on CBNG2 is shown below:

```
supervisor@rtbrick>cbng2.rtbrick.net: cfg> show config link-aggregation
{
  "rtbrick-config:link-aggregation": {
    "interface": [
      {
        "interface-name": "lag-1",
        "mode": "lacp",
        "minimum-link-count": 3,
        "redundancy-session-id": 100,
        "system-id": "a0:aa:aa:aa:aa:aa",
        "member-interface": [
          {
            "member-interface-name": "ifp-0/1/30"
          },
          {
            "member-interface-name": "ifp-0/1/32"
          },
          {
            "member-interface-name": "ifp-0/1/33"
          }
        ]
      },
      {
        "interface-name": "lag-11",
        "mode": "lacp",
        "minimum-link-count": 3,
        "redundancy-session-id": 200,
        "system-id": "a0:aa:aa:aa:aa:a0",
        "member-interface": [
          {
            "member-interface-name": "ifp-0/1/34"
          },
          {
            "member-interface-name": "ifp-0/1/35"
          },
          {
```

```
            "member-interface-name": "ifp-0/1/47"
          }
        ]
      },
      {
        "interface-name": "lag-2",
        "mode": "lacp",
        "minimum-link-count": 1,
        "member-interface": [
          {
            "member-interface-name": "ifp-0/1/14"
          },
          {
            "member-interface-name": "ifp-0/1/15"
          }
        ]
      },
      {
        "interface-name": "lag-3",
        "mode": "lacp",
        "minimum-link-count": 1,
        "member-interface": [
          {
            "member-interface-name": "ifp-0/1/25"
          },
          {
            "member-interface-name": "ifp-0/1/26"
          }
        ]
      }
    ]
  }
}
```

Access Configuration for Redundancy

Run the following commands to configure access interfaces on both of the devices (CBNG1 and CBNG2) for enabling redundancy.

```
set access chassis-id 1
set access aaa-profile ipoe-aaa
set access aaa-profile ipoe-aaa session-timeout 0
set access aaa-profile ipoe-aaa idle-timeout 0
set access aaa-profile ipoe-aaa aaa-radius-profile aaa-radius1
set access aaa-profile ipoe-aaa authentication order RADIUS
set access aaa-profile ipoe-aaa accounting order RADIUS
set access aaa-profile ipoe-aaa accounting interim-interval 30
set access aaa-profile ipoe-aaa accounting session-id-format DEFAULT
set access radius-profile aaa-radius1
set access radius-profile aaa-radius1 nas-identifier 192.0.2.1
```

```
set access radius-profile aaa-radius1 nas-port-type Ethernet
set access radius-profile aaa-radius1 authentication radius-server-profile-name radius-srv1
set access radius-profile aaa-radius1 accounting radius-server-profile-name radius-srv1
set access radius-server radius-srv1
set access radius-server radius-srv1 address 198.51.100.98
set access radius-server radius-srv1 source-address 192.0.2.1
set access radius-server radius-srv1 secret-encrypted-text $2b2feb12f730107454b1be6a0f8242b0f
set access radius-server radius-srv1 routing-instance default
set access radius-server radius-srv1 authentication enable true
set access radius-server radius-srv1 authentication timeout 10
set access radius-server radius-srv1 accounting enable true
set access radius-server radius-srv1 accounting timeout 10
set access radius-server radius-srv1 coa enable true
set access access-profile ipoe
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 address-family ipv4 enable true
set access access-profile ipoe address-family ipv4 pool-name pool1
set access access-profile ipoe address-family ipv4 instance default
set access access-profile ipoe address-family ipv6 enable true
set access access-profile ipoe address-family ipv6 pool-name pool1
set access access-profile ipoe address-family ipv6 prefix-delegation-pool-name pool2
set access access-profile ipoe address-family ipv6 instance default
set access interface double-tagged lag-1 1001 2000 1001 1100
set access interface double-tagged lag-1 1001 2000 1001 1100 access-type IPoE
set access interface double-tagged lag-1 1001 2000 1001 1100 access-profile-name ipoe
set access interface double-tagged lag-1 1001 2000 1001 1100 aaa-profile-name ipoe-aaa
set access interface double-tagged lag-1 1001 2000 1001 1100 gateway-ifl lo-0/0/0/10
set access interface double-tagged lag-1 1001 2000 1001 1100 redundancy-session-id 100
set access interface double-tagged lag-11 2001 3000 1001 1100
set access interface double-tagged lag-11 2001 3000 1001 1100 access-type IPoE
set access interface double-tagged lag-11 2001 3000 1001 1100 access-profile-name ipoe
set access interface double-tagged lag-11 2001 3000 1001 1100 aaa-profile-name ipoe-aaa
set access interface double-tagged lag-11 2001 3000 1001 1100 gateway-ifl lo-0/0/0/10
set access interface double-tagged lag-11 2001 3000 1001 1100 redundancy-session-id 200
set access pool pool1
set access pool pool1 ipv4-address low 198.50.128.1
set access pool pool1 ipv4-address high 198.50.128.255
set access pool pool1 ipv6-prefix low fc55:100:1:1::1/128
set access pool pool1 ipv6-prefix high fc55:100:1:2::5555/128
set access pool pool2
set access pool pool2 ipv6-prefix low fc56:100:1::/56
set access pool pool2 ipv6-prefix high fc56:100:5000::/56set redundancy profile rd_ipoe_olt1
```

Access Configuration for Redundancy on CBNG-1 is shown below:

```
supervisor@rtbrick>cbng1.rtbrick.net: cfg> show config access
{
  "rtbrick-config:access": {
    "chassis-id": 1,
```

```
    "aaa-profile": [
      {
        "profile-name": "ipoe-aaa",
        "session-timeout": 0,
        "idle-timeout": 0,
        "aaa-radius-profile": "aaa-radius1",
        "authentication": {
          "order": "RADIUS"
        },
        "accounting": {
          "order": "RADIUS",
          "interim-interval": 30,
          "session-id-format": "DEFAULT"
        }
      }
    ],
    "radius-profile": [
      {
        "profile-name": "aaa-radius1",
        "nas-identifier": "192.0.2.1",
        "nas-port-type": "Ethernet",
        "authentication": {
          "radius-server-profile-name": [
            "radius-srv1"
            ]
        },
        "accounting": {
          "radius-server-profile-name": [
            "radius-srv1"
            ]
        }
      }
    ],
    "radius-server": [
      {
        "server-name": "radius-srv1",
        "address": "198.51.100.98",
        "source-address": "192.0.2.1",
        "secret-encrypted-text": "$2b2feb12f730107454b1be6a0f8242b0f",
        "routing-instance": "default",
        "authentication": {
          "enable": "true",
          "timeout": 10
        },
        "accounting": {
          "enable": "true",
          "timeout": 10
        },
        "coa": {
          "enable": "true"
        }
```

```
      }
    ],
    "access-profile": [
      {
        "profile-name": "ipoe",
        "protocol": {
          "dhcp": {
            "enable": "true",
            "lease-time": 3600
          },
          "dhcpv6": {
            "enable": "true",
            "lifetime": 3600
          }
        },
        "address-family": {
          "ipv4": {
            "enable": "true",
            "pool-name": "pool1",
            "instance": "default"
          },
          "ipv6": {
            "enable": "true",
            "pool-name": "pool1",
            "prefix-delegation-pool-name": "pool2",
            "instance": "default"
          }
        }
      }
    ],
    "interface": {
      "double-tagged": [
        {
          "interface-name": "lag-1",
          "outer-vlan-min": 1001,
          "outer-vlan-max": 2000,
          "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
        },
        {
          "interface-name": "lag-11",
          "outer-vlan-min": 2001,
          "outer-vlan-max": 3000,
          "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": 200
        }
      ]
    },
    "pool": [
      {
        "pool-name": "pool1",
        "ipv4-address": {
          "low": "11.100.128.1",
          "high": "15.100.128.255"
        },
        "ipv6-prefix": {
          "low": "fc55:100:1:1::1/128",
          "high": "fc55:100:1:2::5555/128"
        }
      },
      {
        "pool-name": "pool2",
        "ipv6-prefix": {
          "low": "fc56:100:1::/56",
          "high": "fc56:100:5000::/56"
        }
      }
    ]
  }
}
```

Access Configuration for Redundancy on CBNG-2 is shown below:

```
supervisor@rtbrick>cbng2.rtbrick.net: cfg> show config access
{
  "rtbrick-config:access": {
    "chassis-id": 2,
    "aaa-profile": [
      {
        "profile-name": "ipoe-aaa",
        "session-timeout": 0,
        "idle-timeout": 0,
        "aaa-radius-profile": "aaa-radius1",
        "authentication": {
          "order": "RADIUS"
        },
        "accounting": {
          "order": "RADIUS",
          "interim-interval": 30,
          "session-id-format": "DEFAULT"
        }
```

```
      }
    ],
    "radius-profile": [
      {
        "profile-name": "aaa-radius1",
        "nas-identifier": "192.0.2.2",
        "nas-port-type": "Ethernet",
        "authentication": {
          "radius-server-profile-name": [
            "radius-srv1"
            ]
        },
        "accounting": {
          "radius-server-profile-name": [
            "radius-srv1"
            ]
        }
      }
    ],
    "radius-server": [
      {
        "server-name": "radius-srv1",
        "address": "198.51.100.98",
        "source-address": "192.0.2.2",
        "secret-encrypted-text": "$2b2feb12f730107454b1be6a0f8242b0f",
        "routing-instance": "default",
        "authentication": {
          "enable": "true",
          "timeout": 10
        },
        "accounting": {
          "enable": "true",
          "timeout": 10
        },
        "coa": {
          "enable": "true"
        }
      }
    ],
    "access-profile": [
      {
        "profile-name": "ipoe",
        "protocol": {
          "dhcp": {
            "enable": "true",
            "lease-time": 3600
          },
          "dhcpv6": {
            "enable": "true",
            "lifetime": 3600
          }
```

```
        },
        "address-family": {
          "ipv4": {
            "enable": "true",
            "pool-name": "pool1",
            "instance": "default"
          },
          "ipv6": {
            "enable": "true",
            "pool-name": "pool1",
            "prefix-delegation-pool-name": "pool2",
            "instance": "default"
          }
        }
      }
    ],
    "interface": {
      "double-tagged": [
        {
          "interface-name": "lag-1",
          "outer-vlan-min": 1001,
          "outer-vlan-max": 2000,
          "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
        },
        {
          "interface-name": "lag-11",
          "outer-vlan-min": 2001,
          "outer-vlan-max": 3000,
          "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": 200
        }
      ]
    },
    "pool": [
      {
        "pool-name": "pool1",
        "ipv4-address": {
          "low": "16.100.128.1",
          "high": "20.100.128.255"
        },
```

```
        "ipv6-prefix": {
          "low": "fc66:100:1:1::1/128",
          "high": "fc66:100:1:2::5555/128"
        }
      },
      {
        "pool-name": "pool2",
        "ipv6-prefix": {
          "low": "fc67:100:1::/56",
          "high": "fc67:100:5000::/56"
        }
      }
    ]
  }
}
```

4.5.2. BNG Blaster Configuration

BNG Blaster is an open-source network testing platform for access and routing protocols. It can emulate massive PPPoE and IPoE (DHCP) subscribers, including IPTV, and L2TP (LNS). One can use this platform for end-to-end BNG and non-BNG router testing. For more information about BNG Blaster, see https://github.com/rtbrick/bngblaster

Configure LAG on BNG Blaster for Redundancy Validation

With BNG blaster, you can validate the Multiservice Edge - IPoE with Redundancy solution implementation and the traffic streams. BNG Blaster uses the following configuration file to validate Multiservice Edge - IPoE with Redundancy.

```
{
  "interfaces": {
    "tx-interval": 1,
    "rx-interval": 1,
    "lag": [
      {
        "interface": "lag1",
        "lacp": true,
        "lacp-timeout-short": true
      },
      {
        "interface": "lag11",
        "lacp": true,
        "lacp-timeout-short": true
      }
    ],
    "links": [
      {
        "interface": "SN-5-C1",
        "lag-interface": "lag1"
```

```
      },
      {
        "interface": "SN-6-C1",
        "lag-interface": "lag1"
      },
      {
        "interface": "SN-7-C1",
        "lag-interface": "lag1"
      },
      {
        "interface": "SN-11-C2",
        "lag-interface": "lag1"
      },
      {
        "interface": "SN-12-C2",
        "lag-interface": "lag1"
      },
      {
        "interface": "SN-13-C2",
        "lag-interface": "lag1"
      },
      {
        "interface": "SN-20-C1",
        "lag-interface": "lag11"
      },
      {
        "interface": "SN-21-C1",
        "lag-interface": "lag11"
      },
      {
        "interface": "SN-22-C1",
        "lag-interface": "lag11"
      },
      {
        "interface": "SN-23-C2",
        "lag-interface": "lag11"
      },
      {
        "interface": "SN-24-C2",
        "lag-interface": "lag11"
      },
      {
        "interface": "SN-25-C2",
        "lag-interface": "lag11"
      }
    ],
    "network": [
      {
        "interface": "SN-19-RR",
        "address": "192.0.2.130/30",
        "gateway": "192.0.2.129",
```

```
        "address-ipv6": "2001:db8::130",
        "gateway-ipv6": "2001:db8::129",
        "isis-instance-id": 1,
        "isis-level": 1,
        "ldp-instance-id": 1
      }
    ],
    "access": [
      {
        "interface": "lag1",
        "type": "ipoe",
        "agent-remote-id": "CBNG1.RTBRICK.{session-global}",
        "outer-vlan-min": 1001,
        "outer-vlan-max": 2000,
        "inner-vlan-min": 1001,
        "inner-vlan-max": 1100
      },
      {
        "interface": "lag11",
        "type": "ipoe",
        "agent-remote-id": "CBNG2.RTBRICK.{session-global}",
        "outer-vlan-min": 2001,
        "outer-vlan-max": 3000,
        "inner-vlan-min": 1001,
        "inner-vlan-max": 1100
      }
    ]
  },
  "sessions": {
    "count": 20000,
    "max-outstanding": 800,
    "start-rate": 400,
    "stop-rate": 400
  },
  "access-line": {
    "agent-remote-id": "DEU.RTBRICK.{session-global}",
    "agent-circuit-id": "0.0.0.0/0.0.0.0 eth 0:{session-global}",
    "rate-up": 2000,
    "rate-down": 16384,
    "dsl-type": 5
  },
  "dhcp": {
    "enable": true
  },
  "dhcpv6": {
    "ldra": true,
    "enable": true
  },
  "session-traffic": {
    "ipv4-pps": 1,
    "ipv6-pps": 1,
```

```
    "ipv6pd-pps": 1,
    "autostart": false
  },
  "ldp": [
    {
      "instance-id": 1,
      "lsr-id": "192.0.2.6",
      "keepalive-time": 30,
      "raw-update-file": "/home/supervisor/ldp_1000.ldp"
    }
  ],
  "isis": [
    {
      "instance-id": 1,
      "area": [
        "49.0002/24"
      ],
      "system-id": "0204.0000.0001",
      "router-id": "192.0.2.6",
      "hostname": "BBL-RR",
      "hello-padding": true,
      "teardown-time": 30,
      "external": {
        "mrt-file": "/home/supervisor/isis_1000.mrt",
        "connections": [
          {
            "system-id": "0300.1001.0001",
            "l1-metric": 10
          }
        ]
      }
    }
  ],
  "bgp": [
    {
      "__comment__": "RR1-C1-IPv4",
      "network-interface": "SN-19-RR",
      "local-ipv4-address": "198.51.100.10",
      "peer-ipv4-address": "192.0.2.1",
      "raw-update-file": "/home/supervisor/ipv4_nlri_nh1.bgp",
      "local-as": 4200000001,
      "peer-as": 4200000001
    },
    {
      "__comment__": "RR1-C1-IPv6",
      "network-interface": "SN-19-RR",
      "local-ipv4-address": "198.51.100.20",
      "peer-ipv4-address": "192.0.2.1",
      "raw-update-file": "/home/supervisor/ipv6_nlri_nh1.bgp",
      "local-as": 4200000001,
      "peer-as": 4200000001
```

```
    },
    {
      "__comment__": "RR1-C2-IPv4",
      "network-interface": "SN-19-RR",
      "local-ipv4-address": "198.51.100.10",
      "peer-ipv4-address": "192.0.2.2",
      "raw-update-file": "/home/supervisor/ipv4_nlri_nh1.bgp",
      "local-as": 4200000001,
      "peer-as": 4200000001
    },
    {
      "__comment__": "RR1-C2-IPv6",
      "network-interface": "SN-19-RR",
      "local-ipv4-address": "198.51.100.20",
      "peer-ipv4-address": "192.0.2.2",
      "raw-update-file": "/home/supervisor/ipv6_nlri_nh1.bgp",
      "local-as": 4200000001,
      "peer-as": 4200000001
    },
    {
      "__comment__": "RR2-C1-IPv4",
      "network-interface": "SN-19-RR",
      "local-ipv4-address": "198.51.100.30",
      "peer-ipv4-address": "192.0.2.1",
      "raw-update-file": "/home/supervisor/ipv4_nlri_nh2.bgp",
      "local-as": 4200000001,
      "peer-as": 4200000001
    },
    {
      "__comment__": "RR2-C1-IPv6",
      "network-interface": "SN-19-RR",
      "local-ipv4-address": "198.51.100.40",
      "peer-ipv4-address": "192.0.2.1",
      "raw-update-file": "/home/supervisor/ipv6_nlri_nh2.bgp",
      "local-as": 4200000001,
      "peer-as": 4200000001
    },
    {
      "__comment__": "RR2-C2-IPv4",
      "network-interface": "SN-19-RR",
      "local-ipv4-address": "198.51.100.30",
      "peer-ipv4-address": "192.0.2.2",
      "raw-update-file": "/home/supervisor/ipv4_nlri_nh2.bgp",
      "local-as": 4200000001,
      "peer-as": 4200000001
    },
    {
      "__comment__": "RR2-C2-IPv6",
      "network-interface": "SN-19-RR",
      "local-ipv4-address": "198.51.100.40",
      "peer-ipv4-address": "192.0.2.2",
```

```
      "raw-update-file": "/home/supervisor/ipv6_nlri_nh2.bgp",
      "local-as": 4200000001,
      "peer-as": 4200000001
    }
  ]
}
```

4.5.3. Validating Multiservice Edge Redundancy and Reachability

The following command line string shows how to start a BNG Blaster instance:

```
sudo bngblaster -C <filename> -I -c10
```

Redundancy Session Data and Subscriber States on Active and Standby Nodes

CBNG-1: Before switchover, which is the actual state

```
supervisor@rtbrick>cbng1.rtbrick.net: op> show redundancy session
Instance: default
  Session ID     Interface      Peer           Source         State          Peer state     Keepalive rcvd      Keepalive
sent
  100            lag-1          192.0.2.200    192.0.2.100    active         standby        91                  92
  200            lag-11         192.0.2.20     192.0.2.10     standby        active         91                  91
supervisor@rtbrick>cbng1.rtbrick.net: op>
```

```
supervisor@rtbrick>cbng1.rtbrick.net: op> show subscriber
Subscriber-Id          Interface        VLAN      Type   State
1369375761697341441    lag-1            1001:1001 IPoE   ESTABLISHED
1369375761697341442    lag-1            1001:1002 IPoE   ESTABLISHED
1369375761697341443    lag-1            1001:1003 IPoE   ESTABLISHED
1369375761697341444    lag-1            1001:1004 IPoE   ESTABLISHED
1369375761697341445    lag-1            1001:1005 IPoE   ESTABLISHED
2522297266304188417    lag-11           2001:1001 IPoE   STANDBY
2522297266304188418    lag-11           2001:1002 IPoE   STANDBY
2522297266304188419    lag-11           2001:1003 IPoE   STANDBY
2522297266304188420    lag-11           2001:1004 IPoE   STANDBY
2522297266304188421    lag-11           2001:1005 IPoE   STANDBY
supervisor@rtbrick>cbng1.rtbrick.net: op>
```

CBNG-2: Before switchover, which is the actual state

```
supervisor@rtbrick>cbng2.rtbrick.net: op>  show redundancy session
Instance: default
  Session ID     Interface      Peer           Source         State          Peer state     Keepalive rcvd      Keepalive
sent
  100            lag-1          192.0.2.100    192.0.2.200    standby        active         94                  93
  200            lag-11         192.0.2.10     192.0.2.20     active         standby        93                  93
supervisor@rtbrick>cbng2.rtbrick.net: op>
```

```
supervisor@rtbrick>cbng2.rtbrick.net: op> show subscriber
Subscriber-Id          Interface        VLAN      Type   State
1369375761697341441    lag-1            1001:1001 IPoE   STANDBY
1369375761697341442    lag-1            1001:1002 IPoE   STANDBY
1369375761697341443    lag-1            1001:1003 IPoE   STANDBY
1369375761697341444    lag-1            1001:1004 IPoE   STANDBY
1369375761697341445    lag-1            1001:1005 IPoE   STANDBY
2522297266304188417    lag-11           2001:1001 IPoE   ESTABLISHED
2522297266304188418    lag-11           2001:1002 IPoE   ESTABLISHED
2522297266304188419    lag-11           2001:1003 IPoE   ESTABLISHED
2522297266304188420    lag-11           2001:1004 IPoE   ESTABLISHED
2522297266304188421    lag-11           2001:1005 IPoE   ESTABLISHED
supervisor@rtbrick>cbng2.rtbrick.net: op>
```

Switchover Scenario for Link Failure Between Active Multiservice Edge Node and Access (OLT) Node

After link (connecting OLT1 to CBNG1) is brought down, the state becomes `down` for the RD session 100 and for the same session, the other CBNG becomes `stand-alone`.

CBNG-1:

```
supervisor@rtbrick>cbng1.rtbrick.net: op> show redundancy session
Instance: default
  Session ID     Interface      Peer           Source         State          Peer state     Keepalive rcvd      Keepalive
sent
  100            lag-1          192.0.2.200    192.0.2.100    down           stand-alone    113                 114
  200            lag-11         192.0.2.20     192.0.2.10     standby        active         114                 114
supervisor@rtbrick>cbng1.rtbrick.net: op>
```

```
supervisor@rtbrick>cbng1.rtbrick.net: op> show subscriber
Subscriber-Id          Interface        VLAN      Type   State
1369375761697341441    lag-1            1001:1001 IPoE   STANDBY
1369375761697341442    lag-1            1001:1002 IPoE   STANDBY
1369375761697341443    lag-1            1001:1003 IPoE   STANDBY
1369375761697341444    lag-1            1001:1004 IPoE   STANDBY
1369375761697341445    lag-1            1001:1005 IPoE   STANDBY
2522297266304188417    lag-11           2001:1001 IPoE   STANDBY
2522297266304188418    lag-11           2001:1002 IPoE   STANDBY
2522297266304188419    lag-11           2001:1003 IPoE   STANDBY
2522297266304188420    lag-11           2001:1004 IPoE   STANDBY
2522297266304188421    lag-11           2001:1005 IPoE   STANDBY
supervisor@rtbrick>cbng1.rtbrick.net: op>
```

CBNG-2:

```
supervisor@rtbrick>cbng2.rtbrick.net: op> show redundancy session
Instance: default
  Session ID     Interface      Peer           Source         State          Peer state     Keepalive rcvd      Keepalive
```

```
sent
  100           lag-1         192.0.2.100    192.0.2.200    stand-alone    down          115              114
  200           lag-11        192.0.2.10     192.0.2.20     active         standby       115              115
supervisor@rtbrick>cbng2.rtbrick.net: op>
```

```
supervisor@rtbrick>cbng2.rtbrick.net: op> show subscriber
Subscriber-Id          Interface        VLAN      Type   State
1369375761697341441    lag-1            1001:1001 IPoE   ESTABLISHED
1369375761697341442    lag-1            1001:1002 IPoE   ESTABLISHED
1369375761697341443    lag-1            1001:1003 IPoE   ESTABLISHED
1369375761697341444    lag-1            1001:1004 IPoE   ESTABLISHED
1369375761697341445    lag-1            1001:1005 IPoE   ESTABLISHED
2522297266304188417    lag-11           2001:1001 IPoE   ESTABLISHED
2522297266304188418    lag-11           2001:1002 IPoE   ESTABLISHED
2522297266304188419    lag-11           2001:1003 IPoE   ESTABLISHED
2522297266304188420    lag-11           2001:1004 IPoE   ESTABLISHED
2522297266304188421    lag-11           2001:1005 IPoE   ESTABLISHED
supervisor@rtbrick>cbng2.rtbrick.net: op>
```

Failed OLT link comes up back (before node failure)

CBNG-1:

```
supervisor@rtbrick>cbng1.rtbrick.net: op> show redundancy session
Instance: default
  Session ID     Interface      Peer           Source         State          Peer state     Keepalive rcvd      Keepalive
sent
  100            lag-1          192.0.2.200    192.0.2.100    standby        active         163                 164
  200            lag-11         192.0.2.20     192.0.2.10     standby        active         163                 163
supervisor@rtbrick>cbng1.rtbrick.net: op>
```

```
supervisor@rtbrick>cbng1.rtbrick.net: op> show subscriber
Subscriber-Id          Interface        VLAN      Type   State
1369375761697341441    lag-1            1001:1001 IPoE   STANDBY
1369375761697341442    lag-1            1001:1002 IPoE   STANDBY
1369375761697341443    lag-1            1001:1003 IPoE   STANDBY
1369375761697341444    lag-1            1001:1004 IPoE   STANDBY
1369375761697341445    lag-1            1001:1005 IPoE   STANDBY
2522297266304188417    lag-11           2001:1001 IPoE   STANDBY
2522297266304188418    lag-11           2001:1002 IPoE   STANDBY
2522297266304188419    lag-11           2001:1003 IPoE   STANDBY
2522297266304188420    lag-11           2001:1004 IPoE   STANDBY
2522297266304188421    lag-11           2001:1005 IPoE   STANDBY
supervisor@rtbrick>cbng1.rtbrick.net: op>
```

CBNG-2:

```
supervisor@rtbrick>cbng2.rtbrick.net: op> show redundancy session
Instance: default
```

```
  Session ID     Interface      Peer           Source         State          Peer state     Keepalive rcvd      Keepalive
sent
  100            lag-1          192.0.2.100    192.0.2.200    active         standby        166                 165
  200            lag-11         192.0.2.10     192.0.2.20     active         standby        165                 165
supervisor@rtbrick>cbng2.rtbrick.net: op>
```

```
supervisor@rtbrick>cbng2.rtbrick.net: op> show subscriber
Subscriber-Id          Interface        VLAN      Type   State
1369375761697341441    lag-1            1001:1001 IPoE   ESTABLISHED
1369375761697341442    lag-1            1001:1002 IPoE   ESTABLISHED
1369375761697341443    lag-1            1001:1003 IPoE   ESTABLISHED
1369375761697341444    lag-1            1001:1004 IPoE   ESTABLISHED
1369375761697341445    lag-1            1001:1005 IPoE   ESTABLISHED
2522297266304188417    lag-11           2001:1001 IPoE   ESTABLISHED
2522297266304188418    lag-11           2001:1002 IPoE   ESTABLISHED
2522297266304188419    lag-11           2001:1003 IPoE   ESTABLISHED
2522297266304188420    lag-11           2001:1004 IPoE   ESTABLISHED
2522297266304188421    lag-11           2001:1005 IPoE   ESTABLISHED
supervisor@rtbrick>cbng2.rtbrick.net: op>
```



To revert to the original active-standby mode based on the priority, a manual switchover can be performed using the `switch-over session 100 confirm` command.

Switchover Scenario for Active Multiservice Edge Node Failure

CBNG-2's Session State and Subscriber State After node failure

```
supervisor@rtbrick>cbng2.rtbrick.net: op> show redundancy session
Instance: default
  Session ID     Interface      Peer           Source         State          Peer state     Keepalive rcvd      Keepalive
sent
  100            lag-1          192.0.2.100    192.0.2.200    stand-alone    invalid        194                 196
  200            lag-11         192.0.2.10     192.0.2.20     stand-alone    invalid        193                 196
supervisor@rtbrick>cbng2.rtbrick.net: op>
```

```
supervisor@rtbrick>cbng2.rtbrick.net: op> show subscriber
Subscriber-Id          Interface        VLAN      Type   State
1369375761697341441    lag-1            1001:1001 IPoE   ESTABLISHED
1369375761697341442    lag-1            1001:1002 IPoE   ESTABLISHED
1369375761697341443    lag-1            1001:1003 IPoE   ESTABLISHED
1369375761697341444    lag-1            1001:1004 IPoE   ESTABLISHED
1369375761697341445    lag-1            1001:1005 IPoE   ESTABLISHED
2522297266304188417    lag-11           2001:1001 IPoE   ESTABLISHED
2522297266304188418    lag-11           2001:1002 IPoE   ESTABLISHED
2522297266304188419    lag-11           2001:1003 IPoE   ESTABLISHED
2522297266304188420    lag-11           2001:1004 IPoE   ESTABLISHED
2522297266304188421    lag-11           2001:1005 IPoE   ESTABLISHED
supervisor@rtbrick>cbng2.rtbrick.net: op>
```

4.6. Appendixes

Appendix B: TACACS+ Server Configuration

The TACACS+ server configuration file (`tac_plus.conf`) can be downloaded from here.

- Click here to download the TACACS+ server configuration file.

Appendix B: RADIUS Server Configuration

The RADIUS server configuration files (`radius_config.zip`) can be downloaded from here. The zip archive contains the set of configuration files needed to configure the RADIUS server.

- Click here to download the RADIUS server configuration files.

Appendix C: RBFS Multiservice Edge1 Configuration

The RBFS Multiservice Edge1 configuration file (`ha-multiservice-edge1.json`) can be downloaded from here.

- Click here to download the RBFS Multiservice Edge1 configuration file.

Appendix D: RBFS Multiservice Edge2 Configuration

The RBFS Multiservice Edge configuration file (`ha-multiservice-edge2.json`) can be downloaded from here.

- Click here to download the RBFS Multiservice Edge2 configuration file.

Appendix E: R-1 (Router 1) Configuration

The Router 1 configuration file (`router1.json`) can be downloaded from here.

- Click here to view th the router 1 configuration file.

Appendix F: R-2 (Router 2) Configuration

The Router 2 configuration file (`router2.json`) can be downloaded from here.

- Click here to view the router 2 configuration file.

Appendix G: Edge Router Configuration

The Edge Router configuration file (`edge-router.json`) can be downloaded from here.

- Click here to view the configuration file.

Appendix H: BNG Blaster Configurations

The BNG Blaster configuration file (`blaster.json`) can be downloaded from here.

- Click here to download the BNG Blaster configuration files.

The JSON file (`isis_3node.json`) which is used to simulate R-1, R-2, and RR on BNG Blaster, can be downloaded from here.

- Click here to download the `isis_3node.json` file.

The BNG Blaster configuration file for Redundancy (`redundancy_blaster.json`) can be downloaded from here.

- Click here to download the BNG Blaster configuration files.

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.