rtbrick

RBFS User Guides

Version 24.3.1, 06 May 2024

Table of Contents

1. Overview Guides

1.1. RBFS Overview

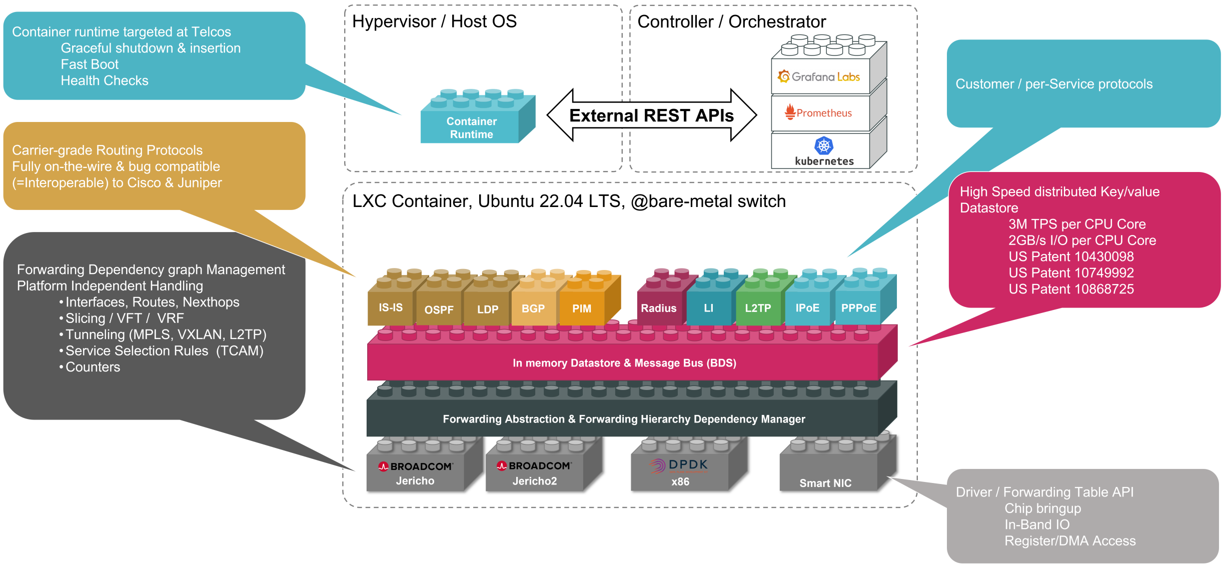
1.1.1. RBFS Overview

RBFS At-a-Glance

RtBrick Full Stack (RBFS) is a disaggregated and open network operating system that is presently productized and available as a Broadband Network Gateway (BNG). RBFS acts as an access software for establishing and managing subscriber sessions for broadband subscribers. It aggregates traffic from various subscriber sessions and routes the traffic to the network of the service provider.

RBFS establishes and maintains a connection with the Customer Premise Equipment (CPE), so that subscribers can access and use the network services from a network service provider.

RBFS runs as an Ubuntu container on the Open Network Linux operating system on white boxes which can perform Layer 2 and Layer 3 switching.


Container runtime targeted at Telcos
Graceful shutdown & insertion
Fast Boot
Health Checks
Carrier-grade Routing Protocols
Fully on-the-wire & bug compatible
(=Interoperable) to Cisco & Juniper
Forwarding Dependency graph Management
Platform Independent Handling
• Interfaces, Routes, Nexthops
• Slicing / VFT / VRF
• Tunneling (MPLS, VXLAN, L2TP)
• Service Selection Rules (TCAM)
• Counters
Hypervisor / Host OS
Container Runtime
External REST APIs
Controller / Orchestrator
Grafana Labs
Prometheus
kubernetes
LXC Container, Ubuntu 22.04 LTS, @bare-metal switch
IS-IS
OSPF
LDP
BGP
PIM
Radius
LI
L2TP
IPoE
PPPoE
In memory Datastore & Message Bus (BDS)
Forwarding Abstraction & Forwarding Hierarchy Dependency Manager
BROADCOM Jericho
BROADCOM Jericho2
DPDK x86
Smart NIC
Customer / per-Service protocols
High Speed distributed Key/value Datastore
3M TPS per CPU Core
2GB/s I/O per CPU Core
US Patent 10430098
US Patent 10749992
US Patent 10868725
Driver / Forwarding Table API
Chip bringup
In-Band IO
Register/DMA Access

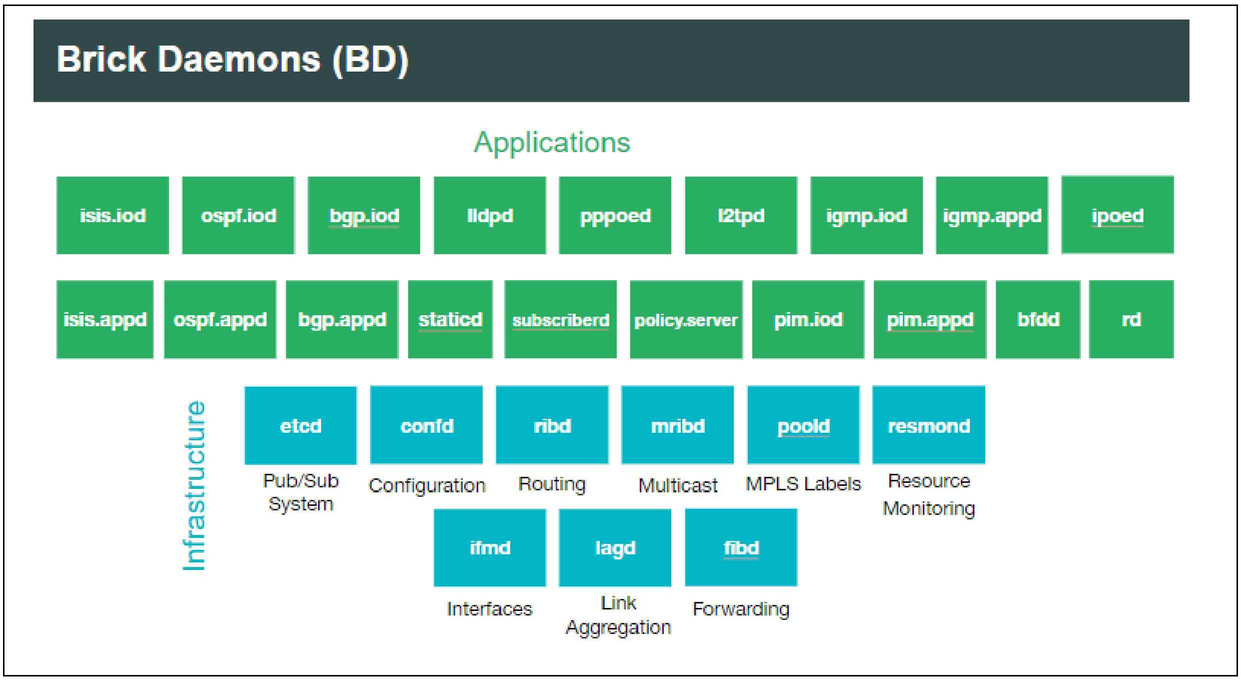
Why RBFS

RBFS' open and disaggregated architectural design fosters a faster deployment of new features and services within a short period and it promotes a collaborative ecosystem of hardware and other component vendors. By separating the hardware from the software, RBFS enables you to choose the white box switches of your choice without any vendor lock-in. It helps to reduce the deployment and operational costs significantly by promoting disaggregated BNG that is suitable for cloud-native ecosystems.

RBFS, built on the microservices architecture, offers some key benefits compared to traditional monolithic systems. It offers greater agility and provides a higher degree of automation that reduces operational overheads. RBFS works well with continuous integration (CI) and continuous delivery (CD) practices and tools.

Architecture and the Key Functional Components

RBFS has been designed based on a microservices architecture to cater a rapidly growing broadband traffic. An RBFS container contains multiple microservices, known as daemons. These microservices are the building blocks of the RBFS ecosystem and they can communicate with each other through a centralized in-memory datastore called Brick Data Store (BDS).



Brick Data Store

RBFS has a schema-driven and in-memory database called BDS (Brick Data Store). As an in-memory data store, BDS relies mainly on the main memory for the storage of data which is contrary to the databases that store data on disks. BDS has architecturally been designed to enable very minimal response time by removing the time to access data stored in disks. BDS acts as a control plane and provides all required data and instructions to the daemons for their functioning.

Brick Daemons

RBFS microservices architecture allows decoupled daemons to serve various functionalities and services and they have their own realm of responsibilities to serve independently.

For example, the subscriber daemon (subscriberd) manages the current subscriber state and is responsible for authentication, authorization, and accounting. The ribd daemon is responsible for route selection, next-hop resolution, tunnel selection and recursion. The forwarding (fibd) daemon handles packet forwarding, route NH download, VPP and PD layer programming. Daemons such as confd and ifmd take care of various configurations and interface management respectively and together they all compose a comprehensive broadband session.

For the routing protocols such as BGP, there are two daemons - bgp.appd and bgp.iod - available to carry out the various functions of the protocol. The bgp.iod daemon manages sending and receiving of the BGP messages such as open, update, keepalive, and notification and takes care of session management. The best route that is selected by bgp.appd daemon is synced with the bgp.iod daemon so that the routes can further be advertised to other BGP peers.

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

In addition to the RtBrick daemons, you can deploy some other third-party applications in the container to bring additional capabilities to the system. For example, Prometheus is an open-source monitoring and alerting software that you can use for observability and monitoring purposes in the container.

Containerization of Daemons

RBFS daemons and other dependencies are packaged as an Ubuntu LXC container. This containerization is a logical layer that helps to make the applications secure, flexible, and portable by providing isolation. This RBFS container is hosted on the

Open Network Linux (ONL), an open-source operating system, which can be run on white box switches.

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

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

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

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

Launching Microservices Dynamically

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

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

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

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

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

Supported Topologies

RBFS can be deployed in a spine-leaf architecture and can also be deployed standalone in a single switch by consolidating all the features in one switch.

A spine-leaf architecture is a two-tier network topology that consists of two switching layers — a spine and a leaf. In this topology, two layers of switches interconnect. The leaf layer consists of access switches that aggregate traffic and connect directly to the spine which is the core network.

The advantage of RBFS spine-leaf topology includes higher performance and better scalability. It is inherently scalable by providing many paths between any two points. This topology is easier for horizontal scaling by adding additional switches to add more capacity to handle increased traffic. This topology is also useful for low latency and higher bandwidth.

A consolidated BNG architecture offers all the functionalities of a spine-leaf BNG architecture on a single bare-metal switch. However, this architecture is recommended when there is a small concentration of broadband subscribers.

Interfaces to Operate and Manage RBFS

RBFS provides a CLI and a rich set of commands that you can use to operate, configure, monitor, and manage the system and its various components. Using the RBFS CLI, you can configure static IPv4, IPv6, MPLS, and multicast routes.

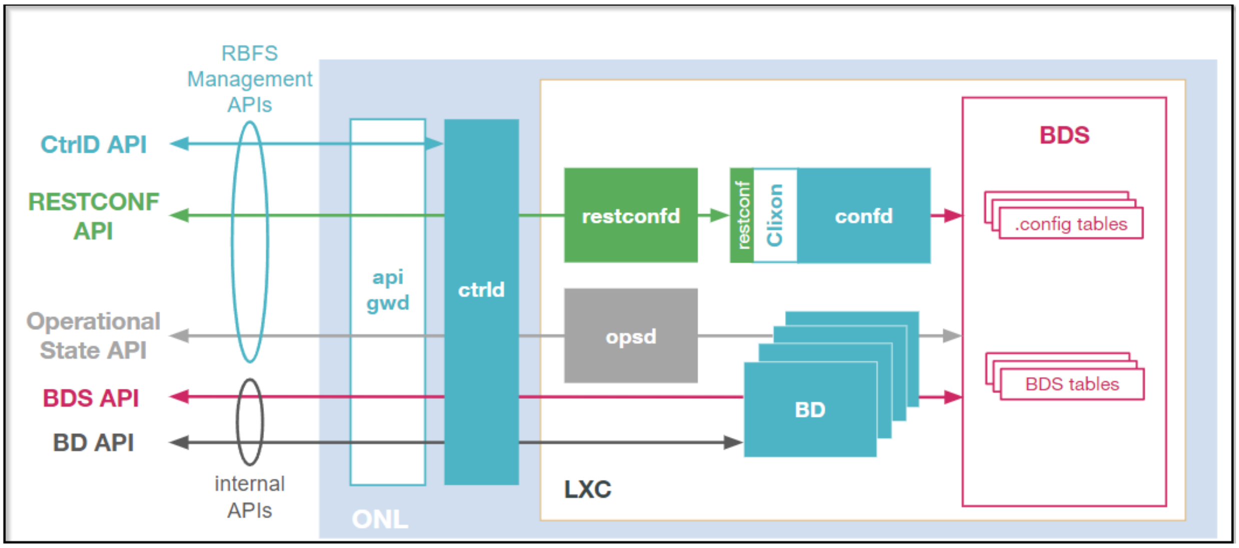
In addition to the CLI, RBFS also offers industry-standard tools and utilities such as RESTCONF.

RBFS supports REST-based industry-standard tools such as RESTCONF and Operational State API to enable communication with the software and underlying devices. RESTCONF is a programmatic interface that enables you to programmatically access RBFS devices and manage configurations.

The Operational State API daemon (opsd) provides the operation state of the system. It forms a stable contract between RBFS and network management systems and inspects the operational state of the device to diagnose and troubleshoot problems.

RBFS APIs allow to access and consume RBFS data simply and securely.

RBMS (RtBrick’s Management System) is a GUI-based application that acts as a single pane of glass and allows interactions with RBFS for all operations, from provisioning and management to monitoring and debugging.



Features and Components

Routing

RBFS, at its core, is a routing software that supports both IP routing and MPLS routing. In dynamic IP routing, RBFS supports all major routing protocols that include OSPFv2 and IS-IS (interior gateway protocols) and BGP (exterior gateway protocol).

RBFS also supports Protocol Independent Multicast (PIM), a multicast routing protocol that runs over existing unicast infrastructure. PIM-SSM uses a subset of PIM sparse mode and IGMP to permit a client to receive multicast traffic directly from the source.

Static Routing

RBFS supports static routing that allows you to configure routes manually.

Segment Routing

RBFS supports segment routing using the IS-IS and OSPF protocols. In segment routing, the source router decides the path (throughout the network) to the destination and encodes the path details in the packet header as an ordered list of instructions. The routers on the path do not take any forwarding decisions but just execute the forwarding instructions.

Routing Policy

RBFS routing policies allow to control and modify the behavior of routing protocols such as IS-IS, OSPF, and BGP. RBFS has a generic routing policy framework that serves multiple purposes and applications. In RBFS, the routing policy implementation is performed by four major components: Policy Repository, Command Processing Module, Policy Server, and Policy Client.

Access and Subscriber Management

RtBrick’s modular and scalable subscriber management offers the next-generation access infrastructure (ng-access) that supports protocols such as PPPoE, IPoE, L2TPv2, DHCPv4 and DHCPv6 and RADIUS. It provides subscriber authentication, access, service creation, activation, and deactivation. It collects accounting statistics for the subscriber sessions. RBFS enables you to address the challenges such as interoperability with numerous client devices from various vendors which requires a well-implemented and industry-proven access protocol stack, including support for all relevant RFCs. RBFS subscriber management infrastructure provides the next generation of internet access protocols designed for carrier-grade services.

Support for PPPoE, IPoE, and L2TPv2

RBFS supports subscriber session management protocols such as Point-to-Point Protocol over Ethernet (PPPoE), Layer Two Tunneling Protocol (L2TPv2), and IP over Ethernet (IPoE) to deliver network access services to broadband subscribers.

PPPoE establishes a PPP connection over the ethernet. In RBFS, the PPPoE daemon (pppoed) manages PPPoE and PPP sessions.

IP-over-Ethernet (IPoE) is an alternative to PPPoE to deliver network access services to broadband subscribers. IPoE does not require client dial-in software and is easy to use when accessing the network. In RBFS, the IPoE daemon (ipoed) manages IPoE services using DHCPv4 and DHCPv6.

The L2TPv2 daemon (l2tpd) is used for the L2TPv2 tunnel and session handling. L2TP is a Layer-3 tunneling protocol that initiates a tunnel between an L2TP access concentrator (LAC) and an L2TP network server (LNS). This enables Point-to-Point Protocol (PPP) link layer to be encapsulated and transferred across the internet.

Accounting

RBFS accounting is the process of tracking subscriber activities and network usage in a subscriber session for auditing and billing. Accounting tracks information such as subscriber identity, the number of packets and bytes transferred from and to the network, start and stop times of the sessions and so on. The accounting keeps track of resources used by the subscriber during the sessions. This includes the session time called time accounting and the number of packets and bytes transmitted during the session called volume accounting. In RBFS, accounting can be performed based on classes or types of services such as video, VoIP, and data.

Support for Lawful Interception

RBFS supports Lawful Interception (LI) to allow legal authorities to obtain communications network data for analysis or evidence. LI is a technique of intercepting certain user data streams tunneling the intercepted traffic to a mediation device with the data and only the users with appropriate credentials can access the intercepted data.

HTTP Redirect Service

RBFS HTTP Redirect service allows network service providers to intercept and redirect HTTP request traffic from subscribers to a designated captive portal instead of the original destination. This powerful service has a multitude of use cases, ranging from subscriber re-authentication to enforcing acceptance of network usage policies. It allows network service providers to re-authenticate subscribers when necessary and ensure that users explicitly accept network usage policies before accessing services. By implementing the RBFS HTTP Redirect Service, network service providers can efficiently manage user access and enforce compliance with network regulations and policies, ultimately enhancing the overall security and user experience within their network environment.

RBFS (Hierarchical) Quality of Service

RBFS Quality of Service (QoS) is a method of prioritizing network traffic for mission-critical applications and high-priority network services such as voice and video. It provides control over a variety of traffic types and ensures that critical data traffic gets sufficient network resources such as bandwidth.

RBFS can perform priority forwarding of data packets throughout the network. For this preferential forwarding, it identifies and classifies the network traffic. So that the critical network packets get sufficient resources. RBFS QoS ensures the required level of service and provides cost benefits to network providers by enabling them to use network resources efficiently.

RBFS also supports Hierarchical Quality of Service (HQoS), a mechanism that allows you to specify Quality of Service (QoS) behavior for different traffic classes. QoS allows classifying services such as voice and video, but using HQoS, you can apply QoS policies to different users, VLANs, logical interfaces, and so on. RBFS employs HQoS by using the mechanisms such as classifier, queuing, scheduler, policer, shaper, and remarking. HQoS provides a higher degree of granularity in traffic management.

RBFS Carrier-Grade Network Address Translation

RBFS is multi-service edge routing software with which you can deliver both CGNAT and BNG functionalities on a single open switch to reduce costs and increase efficiency.

The RBFS CGNAT or NAT444 solution supports Network Address Port Translation (NAPT), which has the potential to conserve IPv4 addresses for service providers. NAPT is an effective method for allowing multiple devices to connect to the Internet using a single public IPv4 address.

The solution can address the IPv4 depletion challenge of service providers. Using the RBFS CGNAT, service providers can serve a large number of subscribers using a limited number of public IPv4 addresses.

RBFS CGNAT solution has some unique characteristics. Both BNG functionalities and CGNAT functionalities can coexist in a single RBFS device. RBFS CGNAT implements NAT in the chipset that allows for the delivery of CGNAT functionality in-line, fully integrated into the packet processing pipeline alongside other functions in the data plane itself, without requiring any additional chipset resources.

RBFS CGNAT supports deterministic NAT mode of address translation, which provides a consistent mapping of private IPv4 addresses with public IPv4 addresses and port ranges. This mode ensures a one-to-one mapping of private IPv4 addresses with public IPv4 addresses, allowing you to specify the private address and its matching public address and port range. The given private IPv4 address is always translated to the same public address.

Ethernet VPN - Virtual Private Wire Service

RBFS Ethernet VPN - Virtual Private Wire Service (EVPN-VPWS) technology provides point-to-point Layer 2 services over an IP or MPLS network. It is based on the EVPN (Ethernet VPN) technology, which extends the BGP (Border Gateway Protocol) to handle MAC addresses and Ethernet segments in addition to IP prefixes.

EVPN-VPWS allows service providers to offer Layer 2 services with better scalability, flexibility, and ease of operation compared to traditional Layer 2 technologies like VPLS (Virtual Private LAN Service). It uses BGP as the control plane protocol to distribute MAC reachability information across the network, enabling efficient MAC learning and forwarding.

RBFS Redundancy

RBFS supports deployment in redundancy mode that protects from link and node failures. Node and link outages that may occur on an RBFS access network can bring down the subscriber services. RBFS Redundancy helps to minimize the impact of these events and to reduce interruptions and downtime by providing a resilient system.

RBFS Redundancy protects subscriber services from various software and hardware 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 goes down, another node can automatically take over the services.

RBFS Redundancy protects subscriber groups using an active standby node cluster model. RBFS Redundancy architecture consists of an active-standby node cluster and one node is active that runs workloads at a time. The peer node, which is identical to the first node, mirrors the concurrent subscriber state data from the peer and takes over workloads in the event of a node or link failure.

Zero Touch Provisioning

By leveraging the Zero Touch Provisioning (ZTP) feature, you can automate many of the RBFS deployment and setup tasks. ZTP allows you to set up and configure the platforms automatically by eliminating the repetitive manual tasks in a large-scale environment. This feature significantly reduces human touch points and errors prone by manual interventions and makes the deployment easier.

Scalability in RBFS

RBFS allows horizontal scaling to enhance system capacity. You can add additional switches to the spine and leaf layers to enhance capacity to handle increased subscriber traffic.

RBFS offers subscriber management capacity in a scale-out architecture called the Point-of-Deployment (PoD), also known as a SEBA PoD (SDN-enabled PoD). A large-scale PoD consists of access leaf routers aggregated by a layer of spine routers in an auto-provisioned CLOS topology. The access leaf routers provide subscriber management functionality. For even greater scalability, a layer of border leaf routers can be added to the core of the network provider network to provide more connectivity.

The leaf routers can be scaled out horizontally to increase the number of subscribers supported on the PoD, providing a pay-as-you-grow model. PPPoE subscribers can be terminated on the access leaf routers or tunneled to an LNS over L2TPv2. L2 Cross Connect (L2X) allows subscriber traffic to be tunneled out of the PoD at Layer 2, providing connectivity.

Security in RBFS

In RBFS, security is integrated into the foundation of the network. RBFS implements several techniques and methods to safeguard the entire network infrastructure. RBFS has a comprehensive set of security capabilities that deploy multiple security controls to protect different areas of the system and network.

Security features for RBFS Control Plane

RBFS Control Plane security feature enables filtering and rate-limiting the traffic transmitted from the forwarding plane to the control plane. RBFS uses Access Control Lists (ACLs) and policers to secure the router’s control plane.

All routing protocols, management protocols, and service protocols run in the control plane. The output of these protocols results in databases such as routing tables, MAC tables, ARP tables, and so on, which eventually get programmed in the forwarding plane.

ACLs are the building blocks of control-plane security. RBFS employs fundamental mechanisms - Protocol ACLs and Route Lookup - for redirecting control plane traffic to the CPU and policers for controlling CP traffic to the CPU.

All routing protocols (BGP, OSPF, and ISIS), Management Protocols (SSH, RESTCONF, and so on), Service Protocols (RADIUS, NTP, and TACACS+), and Access Protocols (PPPoE, DHCP, L2TP, and PPP) automatically create Access Control Lists (ACLs) required to punt the protocol traffic to the CPU Control Plane.

The RBFS Control Plane Security feature adds policers to all protocol ACLs. This feature creates a set of default policers and applies them to the protocol ACLs to secure the control plane from DDoS attacks.

Security features for RBFS Management Plane

RBFS provides the capability to restrict access to the management plane only to authenticated and authorized entities. The authentication identifies the entity and the authorization validates if the entity is allowed to execute the action.

RBFS supports the security protocol, TACACS (Terminal Access Controller Access Control System). RBFS provides a Pluggable Authentication Module (PAM) that enables it to work with TACACS for centralized authentication for users who try to access a router.

For management plane security, RBFS implements token-based authentication that provides access to the management plane through APIs only to the authenticated entities.

RBFS uses JSON web token, an open standard token, that defines a compact and self-contained way for securely transmitting information between parties as a JSON object. The ApiGwD daemon validates the access token against a JSON web key set (JKWS).

Logging and Observability in RBFS

RBFS logging is the process of writing log messages during the execution of an event. Logging provides reports about the events in the entire RBFS ecosystem at different functional areas. You can configure logging based on the different severity levels available. RBFS also allows you to send logs to third-party log management servers such as Graylog where you can view and analyze the real-time data. It provides you the ability to trace out the errors of the applications in real-time.

Operational state visibility is crucial for troubleshooting, testing, monitoring, and capacity management. To enable operational visibility, it is required to collect router metrics periodically. RBFS allows the ingestion of time-series data allows to send operational queries.

RBFS uses Prometheus, an open-source system monitoring and alerting tool, for monitoring and metric collection. Prometheus collects time-stamped data for events, network data, application performance, and so on. The tool allows analyzing metrics with the PromQL query language. Additionally, RBFS provides an optional alert management tool. You can use both of these tools together with its own services to integrate them into the RBFS ecosystem.

Observability Using SNMP

RBFS SNMP (Simple Network Management Protocol) provides a network monitoring mechanism that collects state information from various network devices and components. With SNMP, you can monitor interfaces, CPU usage, temperature of the device, bandwidth usage, and so on. For example, if an interface goes down on one of the devices, SNMP can quickly alert this. The RBFS SNMP implementation allows retrieving system state information using the Protocol Data Unit (PDU) from various network components.

SNMP allows performing various operations that include GET for retrieving data, SET for modifying data, TRAP for notifying an event and so on. These operations provide management access to the MIB hierarchy. RBFS supports the SNMP version 2c and SNMP version 3.

Resource Monitoring

Monitoring the device and its various components is very crucial to analyze the health of devices. RBFS provides resource monitoring capabilities to keep track of various components of the devices. RBFS has a dedicated daemon called resmond to discover and monitor the device resources. With RBFS Resource Monitoring, you can continuously observe the health of the system resources such as CPU, Memory, Processes, Disks, Sensor, and Optics.

Port Mirroring

RBFS supports port mirroring, a monitoring technique that can be implemented on network switches. Port mirroring allows to copy and send data packets from one port to another port for monitoring purposes. Port mirroring enables network administrators to troubleshoot the system with a protocol analyzer on the port that has the mirrored data.

RBFS Software Licensing

RBFS software is available at RtBrick Image Store (https://releases.rtbrick.com/) where you can download the latest version. For more information on RBFS software licensing and installation, see /resources/techdocs/24.3.1/tools/rtb_tools_installation_manual.html[RBFS Software Licensing and Installation].

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

1.2. RBFS CLI Overview

1.2.1. RBFS Command Line Interface

RBFS CLI Overview

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

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

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

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

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

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

Using the CLI

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

Complete Partially Typed Commands:

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

Command Options and Description:

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

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

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

Launch the RBFS CLI

The following example shows how to start the RBFS CLI.

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

> 'op>' shows you are in operational mode.

CLI Prompt

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

The format of the RBFS CLI prompt is as follows:

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

Example:

```
supervisor@rtbrick>LEAF01: op>
```

Switch CLI Modes

RBFS CLI has three modes: Configuration mode, Operation mode, and Debug mode.

You can enter switch-mode command to change the CLI mode.

For example, enter switch-mode config to switch to configuration mode.

The following example shows how to switch between modes.

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

The following example shows how to switch from the operation mode to the config mode.

```
supervisor@rtbrick>LEAF01: op> switch-mode config
supervisor@rtbrick>LEAF01: cfg>
```

Turn on/off Paging

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

paging [**on** | **off**]

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

Example:

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

Display Command History

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

history

Example:

```
supervisor@rtbrick>LEAF01: op> history
show config set
exit
show config set
load config test.json
load config obj.json
show config set
exit
show config set
load config test.json
switch-mode config
load config test.json
load config obj.json
exit
switch-mode config
show config set
load config test.josn
load config obj
load config obj.json
exit
show config set
load config obj.json
load config test.json
exit
show bd running-status
load config test.json
show config set
exit
show bd running-status
show co
show cores
exi
show datastore confd table test index index2
exit
```

CLI Access Logs

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

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

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

System logging is implemented for RESTCONF.

For RESTCONF error logs, do not set the log level to 'info'. If you

> set the log level to info, logs are generated for all the restconfd requests.

Operational Commands

Display Core Files

You can use the show cores command to show a set of system core files created when the device service has been crashed. This command is used for diagnostic purposes.

Example:

```
supervisor@rtbrick>LEAF01: op> show cores
Date    Time    Filename
May 10  09:02   core.lldpd.20220510-090237.1317.zst
May 10  09:02   core.igmp.iod.1.20220510-090228.1282.zst
May 10  09:02   core.pim.iod.1.20220510-090228.1280.zst
May 10  09:01   core.lldpd.20220510-090145.984.zst
May 10  09:01   core.igmp.iod.1.20220510-090140.991.zst
May 10  09:01   core.pim.iod.1.20220510-090140.989.zst
```

View Hardware Resource Usage Limit Information

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

Run the following command:

show hardware limits

Example:

```
supervisor@rtbrick>rtbrick.net: op> show hardware limits
Hardware resources:
  Module: fib
    ASIC                            : q2c
    Role                            : accessleaf
    Model                           : agcva48s
    IPv4 route count                : 1200000
    IPv6 route count                : 250000
  Module: fib
    ASIC                            : q2c
    Role                            : accessleaf
    Model                           : as7946-74xkb
    IPv4 route count                : 1200000
    IPv6 route count                : 250000
  Module: fib
    ASIC                            : q2c
    Role                            : accessleaf
    Model                           : as7946-30xb
    IPv4 route count                : 1200000
    IPv6 route count                : 250000
  Module: fib
    ASIC                            : q2c
    Role                            : accessleaf
    Model                           : s9600-72xc
    IPv4 route count                : 1200000
    IPv6 route count                : 250000
  Module: bgp
    ASIC                            : q2c
    6PE label                       : 2
  Module: confd
    ASIC                            : q2c
    Max MTU profile                 : 8
    Max L3 MTU profile              : 3
    Max subscriber MTU profile      : 6
    Max physical MTU profile        : 8
  Module: rd
    ASIC                            : q2c
```

Reboot Containers and Hosts

The reboot command allows you to restart containers and hosts.

reboot <option>

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

Example:

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

Display System Version Details

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

show version

Example:

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

Display Date and Time

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

date

Example:

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

Display Routes

The show route command displays information of routes.

Syntax:

show route <options>

Attribute	Description
-	Without any option, the command displays the information for all routes for all modules.
detail	Shows detailed route information.
instance <name>	Routing table information for a specified instance.
ipv4	Shows route information for the IPv4 routing table.
ipv6	Shows route information for the IPv6 routing table.
mpls	Shows route information for the MPLS routing table.
label <value>	Shows route information for a specified destination label.
prefix <value>	Shows route information for a specified destination prefix.
prefix-length-distribution	Shows the number of routes with the same prefix length for the sources.
source	Shows routes from a specified source.
summary	Shows the number of routes selected by RIBD for each source.

Example 1: Route information

```
supervisor@rtbrick>LEAF01: op> show route
Instance: default, AFI: ipv4, SAFI: unicast
Prefix/Label        Source            Pref    Next Hop                Interface
11.0.0.1/32         arp-nd            6       11.0.0.1                hostif-0/0/4/1
12.1.0.0/24         ospf              10      23.0.0.2                hostif-0/0/0/1
23.0.0.0/24         direct            0       23.0.0.0                hostif-0/0/0/1
25.0.1.0/24         ospf              10      23.0.0.2                hostif-0/0/0/1
25.1.1.0/24         ospf              10      23.0.0.2                hostif-0/0/0/1
34.0.3.3/32         direct            0       34.0.3.3                hostif-0/0/2/1
56.0.1.4/30         ospf              10      23.0.0.2                hostif-0/0/0/1
56.0.2.0/31         ospf              10      34.0.2.4                hostif-0/0/1/1
```

Example 2: Route summary

```
supervisor@rtbrick>LEAF01: cfg> show route summary
Instance: default
  Source                 Routes
  bgp                         2
  direct                      4
  Total Routes                6
Instance: ip2vrf
  Source                 Routes
  bgp                         6
  direct                      2
  Total Routes                8
Instance: li-vrf
  Source                 Routes
  bgp                         4
  direct                      2
  Total Routes                6
Instance: mgmt-vrf
  Source                 Routes
  bgp                         2
  direct                      2
  Total Routes                4
Instance: radius-vrf
  Source                 Routes
  bgp                         5
  direct                      2
  Total Routes                7
```

Example 3: Routes with the same prefix length for IPv4

```
supervisor@rtbrick>LEAF01: cfg> show route prefix-length-distribution
Instance: default
  Prefix Length      Count
            /32          2
           /128          4
            Sum          6
Instance: ip2vrf
  Prefix Length      Count
             /0          2
            /24          1
            /32          2
            /64          1
           /128          2
            Sum          8
Instance: li-vrf
  Prefix Length      Count
             /0          2
            /32          2
           /128          2
            Sum          6
Instance: mgmt-vrf
  Prefix Length      Count
            /32          2
           /128          2
            Sum          4
Instance: radius-vrf
  Prefix Length      Count
             /0          2
            /24          1
            /32          2
           /128          2
            Sum          7
```

Show Route Resolution

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

show route-resolution <options>

-	**Without any option, the command displays the information for all requests and response tables side by side.**
destination-instance	Displays the information for all requests and response for a destination instance.
look-up instance	Displays lookup instance routes.
prefix	Displays routes for prefix 4 or prefix 6.
resolved	Displays resolved routes.
source	Displays source of requested source.
unresolved	Displays unresolved routes.

Example:

```
supervisor@L1-STD-2-2008>bm08-tst.hel.rtbrick.net: op> show route-resolution
192:1::1, Source: bgp
  Destination instance: default, AFI: ipv4, SAFI: vpn-unicast
  Lookup      instance: default, AFI: ipv6, SAFI: labeled-unicast
  Covering Prefix: 192:1::1/128
    Interface        MAC Address         Nexthop
    hostif-0/0/1/10  7a:11:21:c0:00:03   fe80::7811:21ff:fec0:3
192:1::1, Source: bgp
  Destination instance: default, AFI: ipv4, SAFI: vpn-multicast
  Lookup      instance: default, AFI: ipv6, SAFI: labeled-unicast
  Covering Prefix: 192:1::1/128
    Interface        MAC Address         Nexthop
    hostif-0/0/1/10  7a:11:21:c0:00:03   fe80::7811:21ff:fec0:3
    <...>
```

Example:

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

Viewing Configuration

View Configuration

To view configurations, enter the show config command.

Example:

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

Display Configurations in a Specific Format

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

show config <format>

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

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

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

```
supervisor@rtbrick>LEAF01: op> show config text
daemon-options {
    instance-name *;
    afi *;
    safi *;
    bd-type bgp.appd;
    bd-name bgp.appd.1;
}
daemon-options {
    instance-name *;
    afi *;
    safi *;
    bd-type bgp.iod;
    bd-name bgp.iod.1;
}
interface {
    name lo-0/0/0;
    unit {
        unit-id 0;
        address {
            ipv4 {
                prefix4 198.51.100.75/24;
            }
            ipv6 {
                prefix6 2001:db8:0:110::/32;
            }
        }
    }
}
<...>
```

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

Example:

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

View Configuration in a Specified Hierarchy

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

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

Commit CLI Configurations

To commit the configurations, use the commit command.

The following example shows how to commit your changes.

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

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

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

Add a Configuration Description

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

set system config-description <description>

Example:

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

View Uncommitted Changes

To view the uncommitted changes, use the show diff command:

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

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

```
+interface {
+    name ifp-0/0/4;
+    ifp-id 4;
+}
 instance {
```

Save Configuration

To save configurations, enter the following command:

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

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

Delete the Entire Running Configuration at a Time

To delete the entire running configurations at a time, use the discard all command.

Example:

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

View the Configuration Differences in the Current and Previous Versions

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

show diff <number>

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

Rollback to a Previously Committed Configuration

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

rollback <number>

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

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

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

Rollback to a Specific Version of Software Configuration

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

rollback commit-id <commit-hash>

Example:

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

Load Configuration

To load configurations, enter the following command:

load config <filename> <option>

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

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

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

Discard the Uncommitted Configuration

To discard the uncommitted configuration, enter the following command:

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

Supported Platforms

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

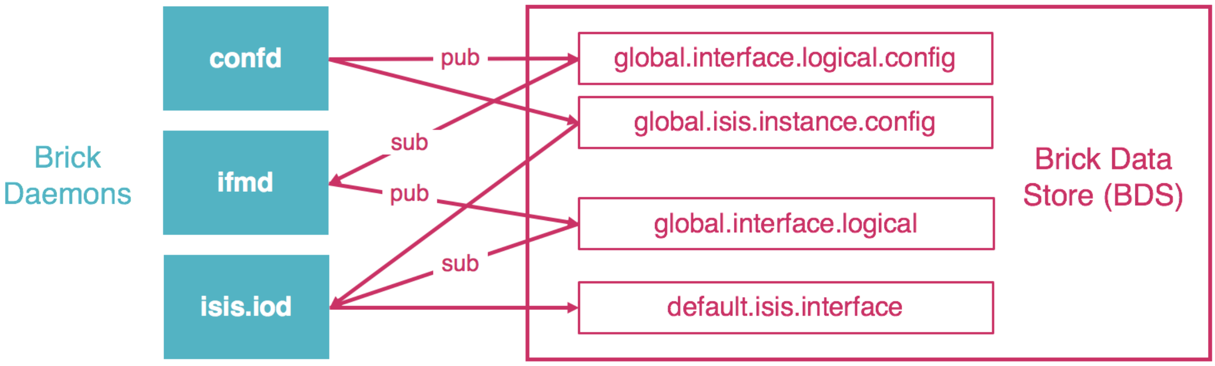
1.3. BDS Overview

1.3.1. BDS Overview

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

Pub/Sub Model

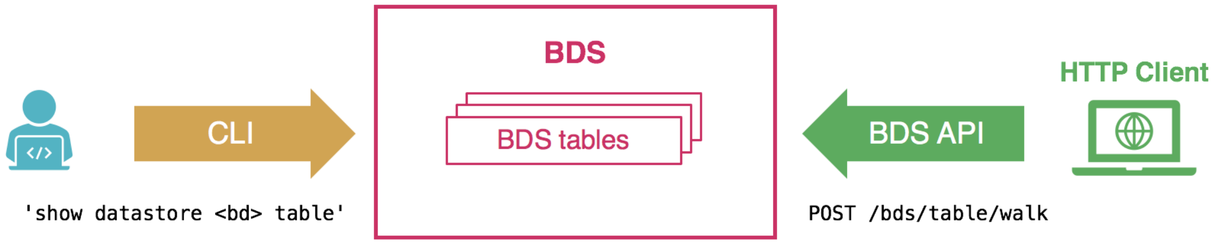
All Brick Daemons (BD) independently publish and subscribe to tables in a pub/sub model. This model provides resilience and scalability. The figure illustrates the concept:



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

BDS User Interface

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



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

Supported Platforms

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

1.3.2. BDS Operational Commands

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

BDS Summary

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

Syntax:

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

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

Example:

```
supervisor@rtbrick>LEAF01: op> show datastore ribd summary
Brick Datastore Summary:
Table Name: local.bds.table.registry.ribd
  Index                                            Type                 Active
Obj Memory   Index Memory
  sequence-index                                   bds_rtb_bplus           234
24.38 KB     9.59 KB
  gc-index                                         bds_rtb_bplus             0   0
bytes      0 bytes
  table-name-index                                 bplus                   234
24.38 KB     9.59 KB
Table Name: local.trim.qrunner.table
  Index                                            Type                 Active
Obj Memory   Index Memory
  sequence-index                                   bds_rtb_bplus             7
840 bytes    728 bytes
  gc-index                                         bds_rtb_bplus             0   0
bytes      0 bytes
  immutable_index                                  bplus                     7
840 bytes    728 bytes
  qrunner-index                                    qrunner                   7
840 bytes    728 bytes
Table Name: local.bds.statistics
  Index                                            Type                 Active
Obj Memory   Index Memory
  sequence-index                                   bds_rtb_bplus           319
34.84 KB     14.36 KB
  gc-index                                         bds_rtb_bplus             0   0
bytes      0 bytes
  immutable-index                                  bplus                   319
34.84 KB     14.36 KB
Table Name: local.bds.module.registry
  Index                                            Type                 Active
Obj Memory   Index Memory
  sequence-index                                   bds_rtb_bplus            68
3.53 KB      2.79 KB
  gc-index                                         bds_rtb_bplus             0   0
bytes      0 bytes
  module-name-index                                bplus                    68
3.53 KB      2.79 KB
```

```
<...>
```

BDS Tables

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

Syntax:

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

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

Example 1: Logical Interface Table

```
supervisor@rtbrick>LEAF01: op> show datastore ifmd table global.interface.logical
Object: 0, Sequence 100125, Last update: Mon Apr 03 13:49:39 GMT +0000 2023
  Attribute                                 Type                           Length
Value
  logical_unit_id (1)                       uint16 (3)                          2
0
  ifl_name (2)                              string (9)                         13
ifl-0/1/31/0
  ifp_name (3)                              string (9)                         11
ifp-0/1/31
  instance (5)                              string (9)                          8
default
  mac_address (8)                           macaddr (22)                        6
e8:c5:7a:8f:76:f2
  ipv4_status (10)                          uint8 (2)                           1
up
  ipv6_status (12)                          uint8 (2)                           1
up
  mpls_mtu (13)                             uint16 (3)                          2
1500
  mpls_status (14)                          uint8 (2)                           1
up
  iso_mtu (15)                              uint16 (3)                          2
1500
  iso_status (16)                           uint8 (2)                           1
down
  admin_status (17)                         uint8 (2)                           1
up
  link_status (18)                          uint8 (2)                           1
up
  ifl_type (19)                             uint8 (2)                           1
Logical Sub interface
  operational_status (24)                   uint8 (2)                           1
up
  ifindex (25)                              uint32 (4)                          4
63745
  instance_id (27)                          uint32 (4)                          4
0
<...>
```

Example 2: Filter IPv6 Route Table by Prefix

```
supervisor@rtbrick>LEAF01: op> show datastore ribd table default.ribd.1.fib-
local.ipv6.unicast
 attribute prefix6 2001:db8::1/128
Object: 0, Sequence 1900002, Last update: Mon Apr 03 13:49:39 GMT +0000 2023
  Attribute                                 Type                           Length
Value
  prefix6 (4)                               ipv6prefix (16)                    17
2001:db8::1/128
  source (11)                               uint8 (2)                           1
direct
  sub_src (12)                              uint8 (2)                           1
Host
  nexthop_key (25)                          payload (8)                        24
```

```
3c86eaebe6617cf61ed96c819cfd63839bd90cc85a533067
  preference (40)                          uint32 (4)                          4
0
  bcm_status (52)                          uint8 (2)                           1
None
  return_code (53)                         uint32 (4)                          4
0
  vpp_status (54)                          uint8 (2)                           1
None
  route_status (55)                        uint32 (4)                          4
|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-
```

Example 3: Filter IPv6 Route Table with Exact Match

```
supervisor@rtbrick>LEAF01: op> show datastore bgp.appd.1 table ip2vrf.bgp.rib-
in.ipv4.unicast.198.51.100.30.198.51.100.25 attribute prefix4 198.51.100.11/24
exact
Object: 0, Sequence: 367772, Last update: Wed May 19 08:05:08 GMT +0000 2021
  Attribute                                Type                           Length
Value
  status (1)                               uint8 (2)                           1
Valid
  recv_path_id (2)                         uint32 (4)                          4
0
  prefix4 (3)                              ipv4prefix (13)                     5
198.51.100.40/24
  rd (5)                                   route-distinguisher (40)            8
198.51.100.100:65001
  source (6)                               uint8 (2)                           1
bgp
  sub_src (7)                              uint8 (2)                           1
Local-Peer
  as_path (9)                              array (7), uint32 (4)              20
[57381, 42708, 1299, 5511, 3215]
  origin (10)                              uint8 (2)                           1
IGP
  peer_type (12)                           uint8 (2)                           1
2
  igp_metric (13)                          uint32 (4)                          4
4294967295
  send_path_id (18)                        uint32 (4)                          4
3238151775
  bgp_nh4 (19)                             ipv4addr (12)                       4
198.51.100.30
  community (24)                           array (7), community (27)           8
['1299:20000', '42708:200']
```

BDS Schema

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

Syntax:

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

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

BDS Statistics Memory

The BDS statistics memory command provides detailed memory usage information.

Syntax:

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

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

2. Routing

2.1. RIB

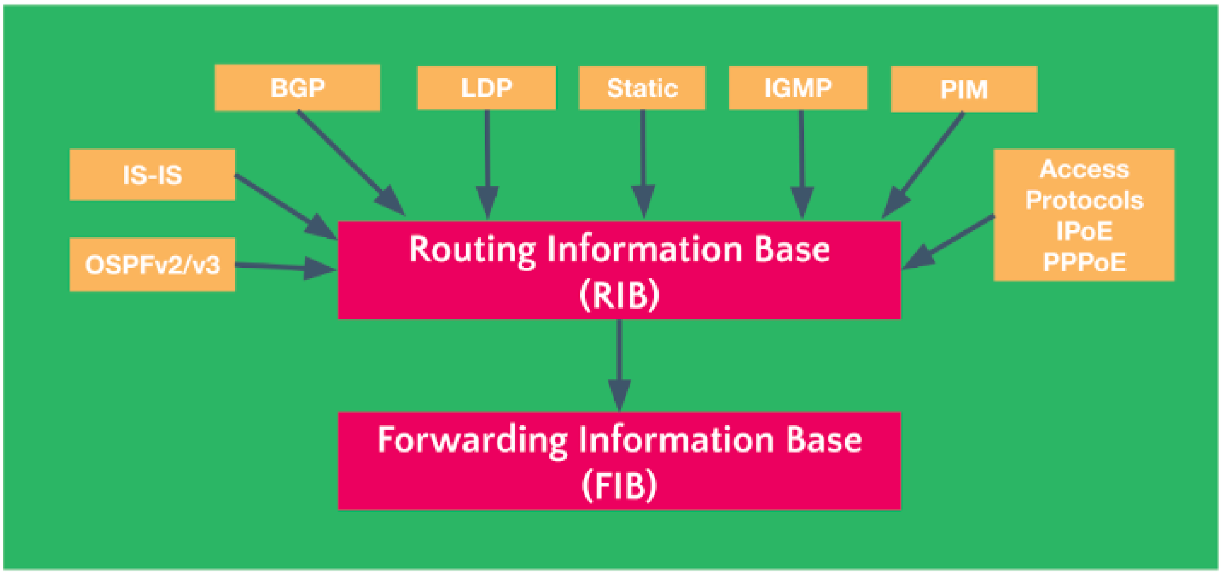
2.1.1. RIB Overview

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

The main differences between RIB and FIB are:

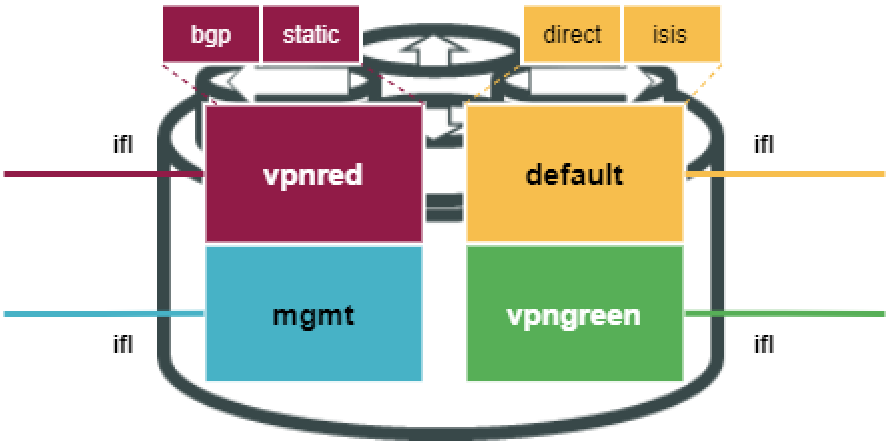
- RIB is about storing and maintaining route information, including attributes for route selection (control plane), etc.
- FIB is used to determine how packets are forwarded (data plane), i.e., only contains a subset of information from RIB. Also, FIB is downloaded to ASIC or VPP.

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


BGP
LDP
Static
IGMP
PIM
IS-IS
OSPFv2/v3
Access Protocols IPoE PPPoE
Routing Information Base (RIB)
Forwarding Information Base (FIB)

Routing Instances

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



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

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

Route Preference

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

Routing Protocol	Route Preference
Direct	0
Static	2
IPoE	7

Routing Protocol	Route Preference
PPP	8
LDP	9
OSPF	10
IS-IS LEVEL 1	15
IS-IS LEVEL 2	18
eBGP	20
iBGP	200

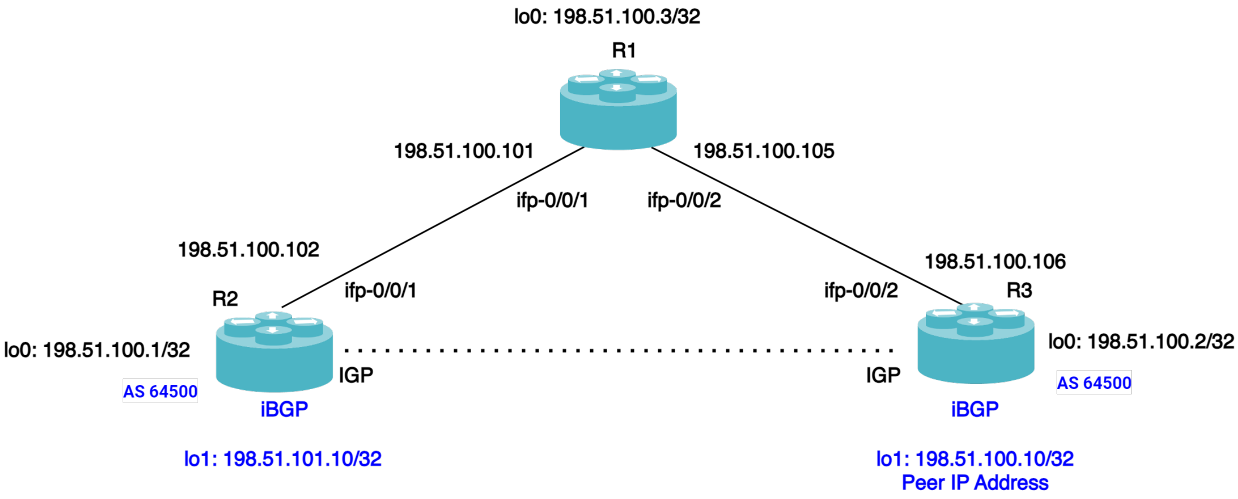
Route Selection

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

Route Resolution

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

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



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

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

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

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

```
supervisor@rtbrick>C-BNG.rtbrick.net: op> show route-resolution resolved prefix
198.51.100.10
198.51.100.10, Source: bgp
  Destination instance: default, AFI: ipv4, SAFI: unicast
  Lookup      instance: default, AFI: ipv4, SAFI: unicast
  Covering Prefix: 198.51.100.10
    Interface        MAC Address         Nexthop
    ifl-0/0/1/13    e8:c5:7a:8f:56:47    198.51.100.101
```

```
supervisor@rtbrick>C-BNG.rtbrick.net: op>
```

Policy Attachments

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

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

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

Nexthops

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

Adjacency

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

FIB

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

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

2.1.2. RIB Operational commands

Display Routes

The show route command displays information on routes.

Syntax:

show route <option>

Option	Description
-	Without any option, the command displays the information for all routes for all instances.
detail	Shows detailed route information.
instance <name>	Routing table information for a specified instance.
ipv4	Shows route information for the IPv4 routing table.
ipv6	Shows route information for the IPv6 routing table.
mpls	Shows route information for the MPLS routing table.
label <value>	Shows route information for a specified destination label.
prefix <value>	Shows route information for a specified destination prefix.
prefix-length-distribution	Shows the number of routes with the same prefix length for the sources.
source	Shows routes from a specified source.
summary	Shows the number of routes selected by RIBD for each source.

Example 1: The following example shows route information.

```
supervisor@rtbrick>LEAF01: op> show route
Instance: default, AFI: ipv4, SAFI: unicast
Prefix/Label                             Source            Pref    Next Hop
Interface
192.1.0.3/32                             direct            0       192.1.0.3
lo-0/0/0/1
```

```
Instance: default, AFI: ipv4, SAFI: labeled-unicast
Prefix/Label                             Source            Pref    Next Hop
Interface                      Label
192.1.0.3/32                             direct            0       192.1.0.3
lo-0/0/0/1                     -
Instance: default, AFI: ipv6, SAFI: unicast
Prefix/Label                             Source            Pref    Next Hop
Interface

192:1::2/128                             bgp               20      fe80::eac5:7aff:fe8f:5663
ifl-0/1/70/13
192:1::3/128                             direct            0       192:1::3
lo-0/0/0/1

192:1::4/128                             bgp               20      fe80::eac5:7aff:fe8f:5663
ifl-0/1/70/13
Instance: default, AFI: ipv6, SAFI: labeled-unicast
Prefix/Label                             Source            Pref    Next Hop
Interface                      Label
192:1::2/128                             bgp               20      fe80::eac5:7aff:fe8f:5663
ifl-0/1/70/13                  -
192:1::3/128                             direct            0       192:1::3
lo-0/0/0/1                     -
192:1::4/128                             bgp               20      fe80::eac5:7aff:fe8f:5663
ifl-0/1/70/13                  2020
```

Example 2: A route summary is shown in the following example.

```
supervisor@rtbrick>LEAF01: op>  show route summary
Instance: default
  Source                 Routes
  bgp                         4
  direct                      4
  Total Routes                8
Instance: inband-vrf
  Source                 Routes
  bgp                         4
  direct                      2
  Total Routes                6
Instance: ip2vrf
  Source                 Routes
  bgp                         3
  direct                      2
  Total Routes                5
Instance: li-vrf
  Source                 Routes
  bgp                         2
  direct                      2
  Total Routes                4
Instance: mgmt-vrf
  Source                 Routes
  bgp                         4
  direct                      2
  Total Routes                6
Instance: radius-vrf
  Source                 Routes
  bgp                         3
  direct                      2
  Total Routes                5
```

Example 3: This example shows routes with IPv4 prefixes of the same length

```
supervisor@rtbrick>LEAF01: op>> show route prefix-length-distribution
Instance: default
  Prefix Length      Count
            /32          2
           /128          6
            Sum          8
Instance: inband-vrf
  Prefix Length      Count
            /32          3
           /128          3
            Sum          6
Instance: ip2vrf
  Prefix Length      Count
            /24          1
            /32          2
           /128          2
            Sum          5
Instance: li-vrf
  Prefix Length      Count
            /32          2
           /128          2
            Sum          4
Instance: mgmt-vrf
  Prefix Length      Count
            /32          3
           /128          3
            Sum          6
Instance: radius-vrf
  Prefix Length      Count
            /24          1
            /32          2
           /128          2
            Sum          5
```

Displaying Route Resolution

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

Syntax:

show route-resolution <options>

Option	Description
-	Without any option, the command displays the information for all requests and response tables side by side.
destination-instance	Displays the information for all requests and responses for a destination instance.

Option	Description
look-up instance	Displays lookup instance routes.
prefix	Displays routes for prefix 4 or prefix 6.
resolved	Displays resolved routes.
source	Displays source of requested source.
unresolved	Displays unresolved routes.

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

```
supervisor@rtbrick>LEAF01: op> show route-resolution
192:1::2, Source: bgp
  Destination instance: default, AFI: ipv4, SAFI: vpn-multicast
  Lookup      instance: default, AFI: ipv6, SAFI: labeled-unicast
  Covering Prefix: 192:1::2/128
    Interface        MAC Address         Nexthop
    ifl-0/1/70/13    e8:c5:7a:8f:56:63   fe80::eac5:7aff:fe8f:5663
fe80::eac5:7aff:fe8f:5663, Source: bgp
  Destination instance: default, AFI: ipv6, SAFI: unicast
  Lookup      instance: default, AFI: ipv6, SAFI: unicast
  Covering Prefix: fe80::eac5:7aff:fe8f:5663/128
    Interface        MAC Address         Nexthop
    ifl-0/1/70/13    e8:c5:7a:8f:56:63   fe80::eac5:7aff:fe8f:5663
fe80::eac5:7aff:fe8f:5663, Source: bgp
  Destination instance: default, AFI: ipv4, SAFI: vpn-unicast
  Lookup      instance: default, AFI: ipv6, SAFI: labeled-unicast
  Covering Prefix: fe80::eac5:7aff:fe8f:5663/128
    Interface        MAC Address         Nexthop
    ifl-0/1/70/13    e8:c5:7a:8f:56:63   fe80::eac5:7aff:fe8f:5663
fe80::eac5:7aff:fe8f:5663, Source: bgp
  Destination instance: default, AFI: ipv6, SAFI: labeled-unicast
  Lookup      instance: default, AFI: ipv6, SAFI: labeled-unicast
  Covering Prefix: fe80::eac5:7aff:fe8f:5663/128
    Interface        MAC Address         Nexthop
    ifl-0/1/70/13    e8:c5:7a:8f:56:63   fe80::eac5:7aff:fe8f:5663
fe80::eac5:7aff:fe8f:5663, Source: bgp
  Destination instance: default, AFI: ipv6, SAFI: vpn-unicast
  Lookup      instance: default, AFI: ipv6, SAFI: labeled-unicast
  Covering Prefix: fe80::eac5:7aff:fe8f:5663/128
    Interface        MAC Address         Nexthop
    ifl-0/1/70/13    e8:c5:7a:8f:56:63   fe80::eac5:7aff:fe8f:5663
```

2.2. BGP

2.2.1. BGP Overview

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

Supported BGP Standards

RFC Number	Description
RFC 2545	Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing
RFC 2918	Route Refresh Capability for BGP-4
RFC 4271	A Border Gateway Protocol 4 (BGP-4)
RFC 4364	BGP/MPLS IP Virtual Private Networks (VPNs)
RFC 4456	BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)
RFC 4486	Subcodes for BGP Cease Notification Message
RFC 4760	Multiprotocol Extensions for BGP-4
RFC 5492	Capabilities Advertisement with BGP-4
RFC 6793	BGP Support for Four-Octet Autonomous System (AS) Number Space
RFC 6608	Subcodes for BGP Finite State Machine Error
RFC 6774	Distribution of Diverse BGP Paths [Partial Support]

RFC and draft compliance are partial except as specified.

Supported Platforms

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

Supported BGP Features

The RBFS supports the following BGP functions:

- Basic BGP Protocol
- Multiprotocol extension for BGP

- Multipath for iBGP and eBGP
- Four-byte AS numbers
- Nexthop Self or next-hop unchanged
- Fast external-failover
- Route reflection
- MD5 Authentication
- Route Refresh
- Advanced route refresh
- Route redistribution
- Multihop EBGP
- Route selection flexibility (always compare MED, ignore AS Path, and so on)
- Add path
- Hostname/Domain name
- Dynamic peers
- Community, Extended Community, and Large Community support
- 6PE Support

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

MD5 Authentication

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

IPv6 Provider Edge (6PE)

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

Policies

The Role of a Routing Policy

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

Attachment Points

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

RtBrick supports attaching a BGP routing policy at two levels:

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

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

Policy Processing

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

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

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

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

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

BGP Best Path Selection Algorithm

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

BGP Best Path Selection Algorithm

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

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

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

same. If a path does not have a local preference attribute (for example, it is received from an eBGP peer), then it is considered to have the local preference assigned in the given BGP instance. The show bgp summary command shows the local preference assigned in the system. This can be changed using the set local-preference value.

3. Prefer the route with the shortest AS path, if no route originated. If there is no AS_PATH attribute, then it is assumed to be of length 0. A single AS_SET is considered to be a length of 1.

4. Prefer the path with the lowest origin type, if the AS path length is the same as all the paths. The available three values include IGP, EGP and Incomplete. The lowest value is IGP and the highest value is Incomplete.

5. Prefer the path with the lowest Multi Exit Discriminator (MED), if the original codes are the same. (By default, MED values are only compared when routes are learned from the same AS. This behavior can be changed using the always-compare-med command. By default, the always-compare-med command is enabled. This command allows the MED values to be compared even if they are learned from different ASs. Routes without MED values are treated as if they have a MED value of 0, which is the lowest and, therefore, always the most preferred value.)

6. Prefer external BGP learned routes over internal BGP routes at this point after comparing the route type (internal BGP and external BGP).

7. Prefer the path whose next hop is resolved through the IGP route with the lowest metric.

8. Prefer the length with a shorter CLUSTER length path. If the CLUSTER attribute is not present, the length is assumed to be 0.

9. Prefer the path from the peer with the lowest router ID. For any path with an originator ID attribute, substitute the originator ID for the router ID during router ID comparison.

10. Prefer the lowest peer IP address as the tie-breaker, if the router-id is the same for both sessions. This is for BGP to make route selections in case of multiple peerings are used between the same routers.

11. If add path is enabled, then the same peer might advertise multiple paths for the same prefix. The path with a lower send path ID is preferred.

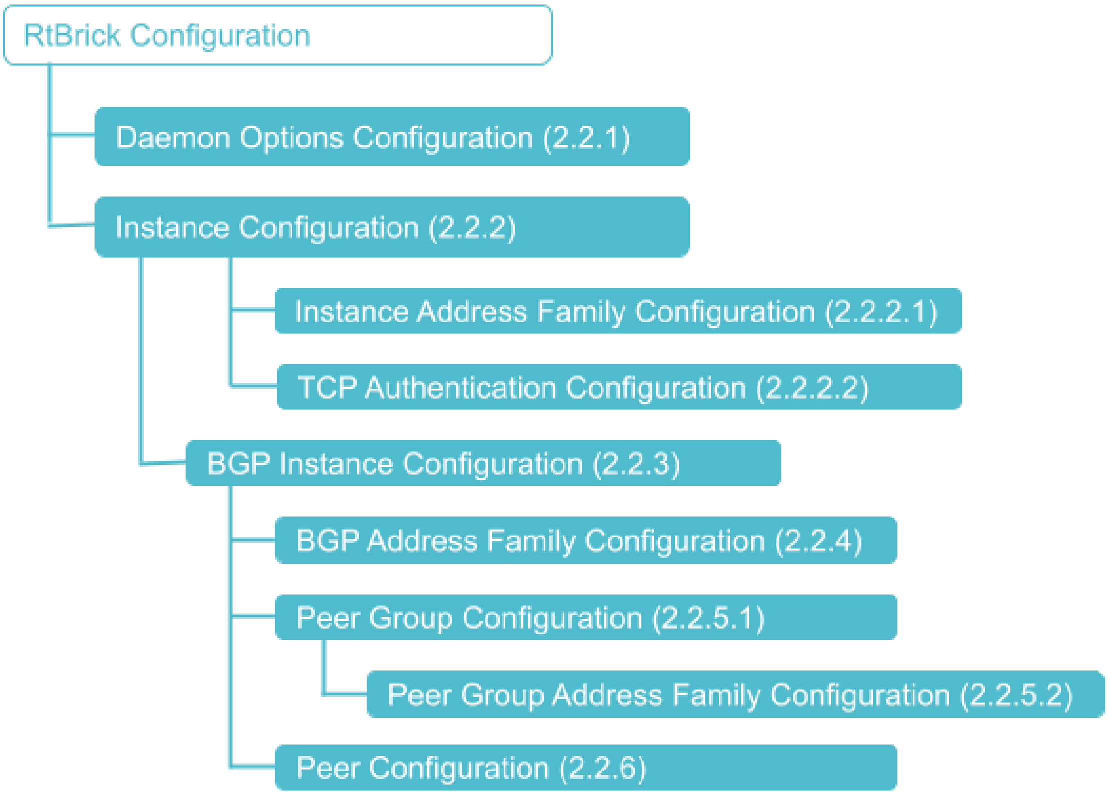
The BGP best path selection algorithm also provides a mechanism to discard paths that are not considered candidates for the best path. The following paths are discarded:

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

2.2.2. BGP Configuration

Configuration Hierarchy

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


RtBrick Configuration
Daemon Options Configuration (2.2.1)
Instance Configuration (2.2.2)
Instance Address Family Configuration (2.2.2.1)
TCP Authentication Configuration (2.2.2.2)
BGP Instance Configuration (2.2.3)
BGP Address Family Configuration (2.2.4)
Peer Group Configuration (2.2.5.1)
Peer Group Address Family Configuration (2.2.5.2)
Peer Configuration (2.2.6)

Configuration Syntax and Commands

The following sections describe the BGP configuration syntax and commands.

Daemon Options Configuration

This configuration associates the BGP daemons with routing instances, AFIs, and SAFIs.

⚠ The BGP daemon option configurations have been deprecated. These will be removed in a subsequent release.

Syntax:

set daemon-options <instance-name> <attribute> <value>

Attribute	Description
<instance-name>	Name of the BGP instance
<afi>	Address family identifier (AFI).
<safi>	Subsequent address family identifier (SAFI).

Attribute	Description
<bd-type>	Daemon type
bd-name <bd-name>	Daemon name

Example: Daemon Configuration

```
{
    "rtbrick-config:daemon-options": [
      {
        "instance-name": "*",
        "afi": "*",
        "safi": "*",
        "bd-type": "bgp.appd",
        "bd-name": "bgp.appd.1"
      },
      {
        "instance-name": "*",
        "afi": "*",
        "safi": "*",
        "bd-type": "bgp.iod",
        "bd-name": "bgp.iod.1"
      }
    ]
  }
```

Instance Configuration

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

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

Syntax:

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

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

Example: Instance Identifier Configuration

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

Address Families

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

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

Syntax:

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

```
  <value>
```

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

Example: Instance Address Family Configuration

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

TCP Authentication Configuration

In the instance TCP authentication hierarchy, you can optionally enable MD5 or HMAC SHA authentication. Authentication is not configured for BGP directly but for the TCP sessions used by BGP. It is necessary to bind authentication to a peer in order for the authentication to work.

BGP TCP authentication is not backward compatible.

Syntax:

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

Attribute	Description
<authentication-id>	Authentication identifier
type <type>	Authentication identifiers such as MD5
key1-id <key1-id>	Key ID1 of the receiver
key1-encrypted-text <key1-encrypted-text>	Encrypted text of key1
key1-plain-text <key1-plain-text>	Plain text of key1
key2-id <key2-id>	Key ID2 of the receiver

Attribute	Description
key2-encrypted-text <key2-encrypted-text>	Encrypted text of key2
key2-plain-text <key2-plain-text>	Plain text of key2

Example: BGP TCP Authentication Configuration

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

BGP Instance Configuration

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

Syntax:

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

Attribute	Description
host-name <host-name>	The name of the BGP host, to a maximum of 64 characters
domain-name <domain-name>	The name of the BGP routing domain, to a maximum of 64 characters

Attribute	Description
enforce-first-as <enable|disable>	By default, the BGP routing process enforces the First AS feature. It discards updates received from an eBGP peer if the peer does not list its own AS number as the first segment in the AS_PATH BGP attribute. Disable the First AS feature to accept updates without the peer’s source AS matching the first AS in the AS_PATH attribute.
local-as <as-number>	The AS number in four-byte format. The numbers allowed are from 1 to 4294967285.
local-preference <preference-value>	The local preference for the BGP protocol. The numbers allowed are from 0 to 4294967285. The local preference is used to select the exit path for an AS.
med <med-value>	The BGP Multi-Exit Discriminator (MED) value. The numbers allowed are from 0 to 4294967285. When an AS has multiple links to another AS, the MED value is used to determine the exit to use to reach the other AS.
protocol-preference (internal | external) <preference-value>	Protocol preference of routes learned by eBGP ('external'), iBGP ('internal'), or both. This preference is used to select routes learned from multiple protocols.
router-id <router-id>	Router identifier in IPv4 format
cluster-id <cluster-identifier>	The cluster ID associates routers in a group within a BGP routing instance. Routers belong to the same cluster if they have the same cluster-ID. The cluster ID is formatted as an IPv4 address.
timer hold-time <seconds>	Hold timer in seconds. The valid range is 5 to 65535.
timer keepalive <seconds>	Keep a live timer in seconds. The valid range is 5 to 65535.
type-of-service cost <low|normal>	ToS cost field (bit 6) for BGP packets
type-of-service delay <low|normal>	ToS delay field (bit 3) for BGP packets

Attribute	Description
type-of-service precedence <precedence>	ToS IP precedence bits (0 - 2) for BGP packets. Valid precedences are critics, flash, flash-override, immediate, internetwork control, precedence, network control, priority, and routine.
type-of-service reliability <high|normal>	ToS reliability field (bit 5) for BGP packets
type-of-service throughput <high|normal>	ToS throughput field (bit 4) for BGP packets

Example: BGP Instance Configuration

The following example shows some global BGP instance configuration attributes. The further BGP configuration like peer groups and peers is shown in the examples in the subsequent sections.

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

BGP Address Family Configuration

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

Syntax:

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

Attribute	Description
<afi>	Address family identifier (AFI). Supported values: ipv4, or ipv6
<safi>	Subsequent address family identifier (SAFI). Supported values: unicast, labeled-unicast, vpn-unicast, multicast, or vpn-multicast
default-information originate <true|false>	Generate and distribute a default route information
download-count <count>	Forward packets over multiple paths, set maximum prefixes to use
multipath <number>	Enable load sharing among multiple BGP paths
retain-route-target (enable|disable)	Retain VPN routes for all route targets, by default this feature is enabled
resolve-nexthop afi <afi>	Address family to resolve the next-hop
resolve-nexthop safi <safi>	Sub-address family to resolve the next-hop
redistribute <source>	Enable the redistribution feature to dynamically inject specific types of routes into the BGP protocol. Supported route sources are direct, igmp, ipoe, isis, ospf, pim, ppp, static.
redistribute <source> policy <policy>	Attach a policy to the redistribution process
srgb base <value>	Segment Routing Global Block (SRGB) start label. The SRGB is the range of label values reserved for segment routing (SR). These values are assigned as segment identifiers (SIDs) to SR-enabled network nodes and have global significance throughout the routing domain. SRGB is supported for labeled unicast only.

Attribute	Description
srgb index <value>	Segment Routing Global Block (SRGB) index
srgb range <value>	Segment Routing Global Block (SRGB) label range

Example 1: BGP Address Family Configuration with Segment Routing

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

Example 2: BGP Address Family Configuration with Redistribution

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

Example 3: BGP Address Family Configuration with Redistribution and Redistribution Policy

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

Peer Group Configuration

Peer Groups

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

Syntax:

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

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

Address Families

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

Syntax:

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

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

Attribute	Description
route-reflect-client <true|false>	Configure this peer as a route reflector client
policy (import | export) <policy-name>	Apply a routing policy to the peer group

Example: BGP Peer Group Configuration

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

Maximum Prefix Limit

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

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

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

You can configure prefix limits for a peer group.

Syntax:

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

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

Example: BGP Maximum Prefix Limit Configuration

```
supervisor@leaf1: cfg> set instance default protocol bgp peer-group v4_100_as
address-family ipv4 unicast prefix-limit count idle-timeout 5
{
  "ietf-restconf:data": {
    "rtbrick-config:daemon-options": [
      {
        "instance-name": "*",
        "afi": "*",
        "safi": "*",
        "bd-type": "bgp.appd",
        "bd-name": [
          "bgp.appd.1"
          ]
      },
      {
        "instance-name": "*",
        "afi": "*",
        "safi": "*",
        "bd-type": "bgp.iod",
        "bd-name": [
          "bgp.iod.1"
          ]
      }
    ],
    "rtbrick-config:interface": [
      {
        "name": "ifl-0/0/0",
        "host-if": "S1-1-S2",
        "unit": [
          {
            "unit-id": 0,
            "instance": "default",
            "address": {
              "ipv4": [
                {
                  "prefix4": "198.51.100.91/24"
                }
              ]
            }
          }
        ]
      },
      {
        "name": "ifl-0/0/1",
        "host-if": "S1-2-S2",
        "unit": [
          {
            "unit-id": 1,
            "address": {
              "ipv4": [
                {
                  "prefix4": "198.51.100.102/24"
                }
              ]
            }
          }
        ]
      },
      {
        "name": "lo-0/0/0",
        "unit": [
          {
            "unit-id": 0,
            "address": {
              "ipv4": [
                {
                  "prefix4": "198.51.100.46/24"
                }
              ],
              "ipv6": [
                {
                  "prefix6": "2001:db8:0:27::/32"
                }
              ]
            }
          }
        ]
      },
      {
        "name": "lo-0/0/1",
        "unit": [
          {
            "unit-id": 1,
            "address": {
              "ipv4": [
                {
                  "prefix4": "198.51.100.111/24"
                }
              ],
              "ipv6": [
                {
                  "prefix6": "2001:db8:0:223::/32"
                }
              ]
            }
          }
        ]
      }
    ],
    "rtbrick-config:instance": [
      {
        "name": "default",
        "address-family": [
          {
            "afi": "ipv4",
            "safi": "labeled-unicast"
          },
          {
            "afi": "ipv4",
            "safi": "unicast"
          },
          {
            "afi": "ipv6",
            "safi": "labeled-unicast"
          },
          {
            "afi": "ipv6",
            "safi": "unicast"
          },
          {
            "afi": "mpls",
            "safi": "unicast"
          }
        ],
        "protocol": {
          "bgp": {
            "local-as": 200,
            "router-id": "198.51.100.111",
            "address-family": [
              {
                "afi": "ipv4",
                "safi": "unicast",
                "redistribute": [
                  {
                    "source": "direct"
                  }
                ]
              }
            ],
            "peer": {
              "ipv4": [
                {
                  "peer-address": "198.51.100.92",
                  "update-source": "198.51.100.91",
                  "peer-group": "v4_100_as"
                }
              ]
            },
            "peer-group": [
              {
                "pg-name": "v4_100_as",
                "local-as": 200,
                "remote-as": 100,
                "address-family": [
                  {
                    "afi": "ipv4",
                    "safi": "unicast",
                    "prefix-limit": {
                      "count": 100,
                      "idle-timeout": 5
                    }
                  }
                ]
              }
            ]
          }
        }
      }
    ]
  }
}
```

Peer Configuration

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

Syntax to configure a BGP peer by address:

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

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

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

Syntax to configure TCP Authentication for BGP peers:

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

Attribute	Description
interface <name>	Enable BGP peer using IPv6 link-local addresses
ipv4 <peer-address>	IPv4 address of a BGP peer
ipv6 <peer-address>	IPv6 address of a BGP peer
<update-source>	Local IP address to be used for the peering
peer-group <peer-group>	Assign the peer to a peer group
deactivate	Deactivate a configured peer
authentication-id <authentication-id>	Authentication identifier

Example 1: BGP peer specified by IP addresses

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

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

Example 2: BGP peer using IPv6 link-local addresses

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

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

Example 3: BGP peer authentication

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

{
  "rtbrick-config:peer": {
    "interface": [
      {
        "name": "ifl-0/0/1/1",
        "authentication-id": "auth1",
        "peer-group": "spine"
      }
    ]
  }
}
```

Sample Configuration

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

```
{
  "ietf-restconf:data": {
    "rtbrick-config:daemon-options": [
      {
        "instance-name": "*",
        "afi": "*",
        "safi": "*",
        "bd-type": "bgp.appd",
        "bd-name": "bgp.appd.1"
      }
    ],
    "rtbrick-config:instance": [
      {
        "name": "default",
        "ipv4-router-id": "198.51.100.51",
        "protocol": {
          "bgp": {
            "domain-name": "rtbrick.com",
            "host-name": "spine1",
            "local-as": 4200000100,
            "address-family": [
              {
                "afi": "ipv4",
                "safi": "vpn-unicast"
              },
              {
                "afi": "ipv6",
                "safi": "labeled-unicast",
                "srgb": {
                  "base": 5000,
                  "range": 1000,
                  "index": 11
                },
                "redistribute": [
                  {
                    "source": "direct"
                  }
                ]
              },
              {
                "afi": "ipv6",
                "safi": "unicast",
                "redistribute": [
                  {
                    "source": "direct"
                  }
                ]
              },
              {
                "afi": "ipv6",
                "safi": "vpn-unicast"
              }
            ],
            "peer": {
              "interface": [
                {
                  "name": "ifl-0/1/1/1",
                  "authentication-id": "auth1",
                  "peer-group": "spine"
                },
                {
                  "name": "ifl-0/2/1/1",
                  "peer-group": "leaf1"
                },
                {
                  "name": "ifl-0/2/2/1",
                  "peer-group": "leaf2"
                }
              ]
            },
            "peer-group": [
              {
                "pg-name": "leaf1",
                "link-local-nexthop-only": "true",
                "remote-as": 4200000201,
                "address-family": [
                  {
                    "afi": "ipv4",
                    "safi": "vpn-unicast",
                    "extended-nexthop": "true",
                    "nexthop-unchanged": "true"
                  },
                  {
                    "afi": "ipv6",
                    "safi": "labeled-unicast"
                  },
                  {
                    "afi": "ipv6",
                    "safi": "unicast"
                  },
                  {
                    "afi": "ipv6",
                    "safi": "vpn-unicast",
                    "nexthop-unchanged": "true"
                  }
                ]
              },
              {
                "pg-name": "leaf2",
                "link-local-nexthop-only": "true",
                "remote-as": 4200000202,
                "address-family": [
                  {
                    "afi": "ipv4",
                    "safi": "vpn-unicast",
                    "extended-nexthop": "true",
                    "nexthop-unchanged": "true"
                  },
                  {
                    "afi": "ipv6",
                    "safi": "labeled-unicast"
                  },
                  {
                    "afi": "ipv6",
                    "safi": "unicast"
                  },
                  {
                    "afi": "ipv6",
                    "safi": "vpn-unicast",
                    "nexthop-unchanged": "true"
                  }
                ]
              },
              {
                "pg-name": "spine",
                "link-local-nexthop-only": "true",
                "remote-as": 4200000100,
                "address-family": [
                  {
                    "afi": "ipv4",
                    "safi": "vpn-unicast",
                    "extended-nexthop": "true"
                  },
                  {
                    "afi": "ipv6",
                    "safi": "labeled-unicast",
                    "nexthop-self": "true"
                  },
                  {
                    "afi": "ipv6",
                    "safi": "unicast",
                    "nexthop-self": "true"
                  },
                  {
                    "afi": "ipv6",
                    "safi": "vpn-unicast"
                  }
                ]
              }
            ]
          }
        }
      }
    ]
  }
}
```

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

```
{
  "ietf-restconf:data": {
    "rtbrick-config:instance": [
      {
        "name": "default",
        "ipv4-router-id": "198.51.100.53",
        "protocol": {
          "bgp": {
            "domain-name": "rtbrick.com",
            "host-name": "leaf1",
            "local-as": 4200000201,
            "address-family": [
              {
                "afi": "ipv4",
                "safi": "vpn-unicast"
              },
              {
                "afi": "ipv6",
                "safi": "labeled-unicast",
                "srgb": {
                  "base": 5000,
                  "range": 1000,
                  "index": 13
                },
                "redistribute": [
                  {
                    "source": "direct"
                  }
                ]
              },
              {
                "afi": "ipv6",
                "safi": "unicast",
                "redistribute": [
                  {
                    "source": "direct"
                  }
                ]
              },
              {
                "afi": "ipv6",
                "safi": "vpn-unicast"
              }
            ],
            "peer": {
              "interface": [
                {
                  "name": "ifl-0/1/1/1"                        "authentication-
id": "auth1",
                  "peer-group": "spine"
                },
                {
                  "name": "ifl-0/1/2/1",
                  "peer-group": "spine"
                }
              ]
            },
            "peer-group": [
              {
                "pg-name": "spine",
                "link-local-nexthop-only": "true",
                "remote-as": 4200000100,
                "address-family": [
                  {
                    "afi": "ipv4",
                    "safi": "vpn-unicast",
                    "extended-nexthop": "true",
                    "update-nexthop": {
                      "ipv6-address": "2001:db8:0:19::"
                    }
                  },
                  {
                    "afi": "ipv6",
                    "safi": "labeled-unicast"
                  },
                  {
                    "afi": "ipv6",
                    "safi": "unicast"
                  },
                  {
                    "afi": "ipv6",
                    "safi": "vpn-unicast",
                    "update-nexthop": {
                      "ipv6-address": "2001:db8:0:19::"
                    }
                  }
                ]
              }
            ]
          }
        }
      },
      {
        "name": "services",
        "ipv4-router-id": "198.51.100.41",
        "route-distinguisher": "198.51.100.41:101",
        "address-family": [
          {
            "afi": "ipv4",
            "safi": "unicast",
            "policy": {
              "export": "MY_V4_POLICY"
            },
            "route-target": {
              "import": "target:198.51.100.70:14",
              "export": "target:198.51.100.70:14"
            }
          },
          {
            "afi": "ipv6",
            "safi": "unicast",
            "policy": {
              "export": "MY_V6_POLICY"
            },
            "route-target": {
              "import": "target:198.51.100.70:16",
              "export": "target:198.51.100.70:16"
            }
          }
        ],
        "protocol": {
          "bgp": {
            "domain-name": "rtbrick.com",
            "host-name": "leaf1",
            "local-as": 65003,
            "address-family": [
              {
                "afi": "ipv4",
                "safi": "unicast",
                "redistribute": [
                  {
                    "source": "direct"
                  },
                  {
                    "source": "ppp"
                  },
                  {
                    "source": "static"
                  }
                ]
              },
              {
                "afi": "ipv6",
                "safi": "unicast",
                "redistribute": [
                  {
                    "source": "direct"
                  },
                  {
                    "source": "ppp"
                  },
                  {
                    "source": "static"
                  }
                ]
              }
            ]
          }
        }
      }
    ]
  }
}
```

2.2.3. BGP Operational Commands

BGP Show Commands

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

BGP Summary

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

Syntax:

show bgp summary <option>

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

Example: BGP summary for the default instance

```
supervisor@rtbrick: op> show bgp summary instance default
Instance: default
  General information
    Hostname: PE1, Domain name:
    Local AS: 1000, Version: 4
    Local preference: 100, Protocol preference: 200
    Router ID: 198.51.100.102, Cluster ID: 198.51.100.102
  Capabilities
    Route refresh: True, AS4: True, Graceful restart: False
  Best route selection
    Always compare MED: False, Ignore as path: False
    Ignore local preference: False, Ignore origin: False
    Ignore MED: False, Ignore route source: False
    Ignore router ID: False, Ignore uptime: True
    Ignore cluster length: False, Ignore peer IP: False
    Route select parameter: 0
  Timers
    Connect retry: 30s, Keepalive: 30s, Holdtime: 90s
  Statistics
    Peers configured: 1, Peers auto discovery: 0
      Peers in idle          : 0
      Peers in connect       : 0
      Peers in active        : 0
      Peers in opensent      : 0
      Peers in openconfirm   : 0
      Peers in established   : 1
```

BGP Peer

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

Syntax:

show bgp peer <option> ...

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

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

Example 1: BGP Peer Summary View

```
supervisor@rtbrick: op> show bgp peer
Instance: default
  Peer                                     Remote AS    State         Up/Down Time
PfxRcvd              PfxSent
  PE2                                      2000         Established
11d:22h:18m:30s            12                   20
Instance: default
  Peer                                     Remote AS    State         Up/Down Time
PfxRcvd              PfxSent
  CE1                                      65535        Established
6d:02h:28m:02s             2                    2
  CE1                                      65535        Established
6d:02h:27m:45s             2                    2
```

Example 2: BGP Peer Detail View

```
supervisor@rtbrick: op> show bgp peer detail
Peer: PE2, Peer IP: 198.51.100.39, Remote AS: 2000, Local: 198.51.100.29, Local
AS: 1000, Any AS: False
  Type: ebgp, State: Established, Uptime: 11d:22h:18m:48s, Reason: Cease, Sub-
Code: Admin shutdown
  Discovered on interface: -
  Last transition: Thu Nov 19 05:33:28 GMT +0000 2020, Flap count: 1
  Peer ID        : 198.51.100.106, Local ID  : 198.51.100.102
  Instance       : default, Peer group: to_pe2
  6PE enabled    : False
  Timer values:
    Peer keepalive : 30s, Local keepalive: 30s
    Peer holddown  : 90s, Local holddown : 90s
    Connect retry  : 30s
  Timers:
    Connect retry timer : 0s
    keepalive timer     : expires in 1s 488011us
    Holddown timer      : expires in 1m 15s 85437us
  NLRIs:
    Sent           : ['ipv6-unicast', 'ipv4-vpn-unicast', 'ipv6-vpn-unicast',
'ipv6-labeled-unicast']
    Received       : ['ipv6-unicast', 'ipv6-labeled-unicast', 'ipv4-vpn-unicast',
'ipv6-vpn-unicast']
    Negotiated     : ['ipv6-unicast', 'ipv6-labeled-unicast', 'ipv4-vpn-unicast',
'ipv6-vpn-unicast']
  Capabilities:
    Addpath sent                  : None
    Addpath received              : None
    Addpath negotiated            : None
    Extended nexthop sent         : ['ipv4-vpn-unicast']
    Extended nexthop received     : ['ipv4-vpn-unicast']
    Extended nexthop negotiated   : ['ipv4-vpn-unicast']
    Capabilities:
      Feature                      Sent            Received        Negotiated
      Route refresh                True            True            True
      4 byte AS                    True            True            True
      Graceful restart             False           False           False
      Link local only              False           False           False
  End of RIB:
    Address family                 Sent                                Received
    IPv4 unicast                   never                               never
    IPv4 labeled-unicast           never                               never
    IPv6 unicast                   Thu Nov 19 05:33:30 GMT +0000 2020  Thu Nov 19
05:33:30 GMT +0000 2020
    IPv6 labeled-unicast           Thu Nov 19 05:33:30 GMT +0000 2020  Thu Nov 19
05:33:30 GMT +0000 2020
    IPv4 VPN-unicast               Thu Nov 19 05:33:30 GMT +0000 2020  Thu Nov 19
05:33:30 GMT +0000 2020
    IPv6 VPN-unicast               Thu Nov 19 05:33:30 GMT +0000 2020  Thu Nov 19
05:33:30 GMT +0000 2020
    IPv4 VPN-multicast             never                               never
  Message stats:
    Session stats:
      Direction   Open         Update       Keepalive    Notify       Route
refresh
      Input       1            38           41196        0            0
      Output      1            22           41207        0            0
    Total stats:
      Input       2            48           44618        1            0
      Output      3            32           44624        0            0
    Route stats:
      Address family                 Received     Sent
      IPv4 unicast                   0            0
      IPv4 labeled-unicast           0            0
      IPv6 unicast                   2            3
      IPv6 labeled-unicast           2            3
      IPv4 VPN-unicast               4            7
      IPv6 VPN-unicast               4            7
      IPv4 multicast                 0            0
      IPv4 VPN-multicast             0            0
<...>
```

Example 3: BGP Peer Statistics

```
supervisor@rtbrick: op> show bgp peer statistics instance default peer PE2
Instance: default
  Peer                           AFI     SAFI               PfxRcvd    PfxSent
  PE2                            ipv4    unicast            0          0
                                 ipv4    labeled-unicast    0          0
                                 ipv6    unicast            2          3
                                 ipv6    labeled-unicast    2          3
                                 ipv4    vpn-unicast        4          7
                                 ipv6    vpn-unicast        4          7
                                 ipv4    multicast          0          0
                                 ipv4    vpn-multicast      0          0
```

BGP Peer Group

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

Syntax:

show bgp peer-group <option> …

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

Example: BGP Peer Group

```
supervisor@rtbrick: op> show bgp peer-group to_pe2
Instance: default
  Peer group name       : to_pe2
    Remote AS           : 2000
    Import rule         : None
    Export rule         : None
    Remove AS           : None
    Nexthop self        : None
    Multipath iBGP      : None
    Multipath eBGP      : None
    Client-to-Client    : None
    Add path            : None
    eBGP multihop       : None
    Hop (TTL)           : None
    Any AS              : None
    Update VPNv4 NH     : None
    Update MVPN NH      : None
```

BGP FIB

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

Syntax:

show bgp fib <option> …

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

Example 1: Summary view of the BGP FIB for IPv6, all SAFIs and all instances

```
supervisor@rtbrick: op> show bgp fib ipv6
Instance: default, AFI: ipv6, SAFI: unicast
  Prefix                                        Preference      Out Label
Next Hop
  2001:db8:0:2::/32                             20              -
198.51.100.39
  2001:db8:0:2::/32                             20              -
198.51.100.39
Instance: services, AFI: ipv6, SAFI: unicast
  Prefix                                        Preference      Out Label
Next Hop
  2001:db8:0:6::/32                             200             -
2001:db8:0:4::
Instance: default, AFI: ipv6, SAFI: labeled-unicast
  Prefix                                        Preference      Out Label
Next Hop
  2001:db8:0:2::/32                             20              2003
```

```
198.51.100.39
  2001:db8:0:2::/32                             20              2003
198.51.100.39
Instance: default, AFI: ipv6, SAFI: vpn-unicast
  Prefix                                        Preference      Out Label
Next Hop
  2001:db8:0:5::/32                             200             20003,bos:1
  2001:db8:0:6::/32                             200             20003,bos:1
  2001:db8:0:8::/32                             200             20003,bos:1
  2001:db8:0:9::/32                             20              20006,bos:1
2001:db8:0:7::
  2001:db8:0:10::/32                            20              20006,bos:1
2001:db8:0:7::
  2001:db8:0:11::/32                            20              20006,bos:1
2001:db8:0:7::
  2001:db8:0:12::/32                            20              20006,bos:1
2001:db8:0:7::
```

Example 2: Detailed view of the BGP FIB for IPv6 VPN unicast routes in the default instances

```
supervisor@rtbrick: op> show bgp fib ipv6 vpn-unicast instance default detail
Instance: default, AFI: ipv6, SAFI: vpn-unicast
  Prefix: 2001:db8:0:5::/32
    Next hop key: 2b38f6f1d2ae56178666d1edcffd18a85fd4509bcac9a21f
    Peer: None, Peer domain: None
    Route source: bgp-local, Send path ID: 405188370, Received path ID: None, Path
hash: None
    As path: None, Originator ID: None, Origin: Incomplete
    Community: None
    Extended community: ['target:198.51.100.93:2']
    Cluster list: None
    IGP metric: None, Local preference: 100, Multi exit discriminator: 0
    Preference: 200, External route: None, Readvertised route: None
    Label: 20003,bos:1, Route up: None
  Prefix: 2001:db8:0:6::/32
    Next hop key: 62b6c375c2ee2cb053bd5482ec1b7df18e271b6e0d37a4b0
    Peer: None, Peer domain: None
    Route source: bgp-local, Send path ID: 2400017309, Received path ID: None,
Path hash: None
    As path: None, Originator ID: None, Origin: Incomplete
    Community: None
    Extended community: ['target:198.51.100.93:2']
    Cluster list: None
    IGP metric: None, Local preference: 100, Multi exit discriminator: None
    Preference: 200, External route: None, Readvertised route: None
    Label: 20003,bos:1, Route up: None
```

BGP RIB-in

This command displays the received routes.

Syntax:

show bgp rib-in <option> …

Option	Description
-	Without any option, the command displays information on the received BGP routing table on all instances in a summary table format.
<afi>	BGP routing table summary for the given address family (AFI), all sub-address families and all instances. Supported AFI values are 'ipv4' and 'ipv6'.
<afi> <safi>	BGP routing table summary for the given address family (AFI) and sub-address family (SAFI), and all instances. Supported SAFI values are 'labeled-unicast', 'unicast', 'vpn-multicast', and 'vpn-unicast'.
<afi> <safi> detail	Detailed list view of the BGP routing table for the given address family (AFI) and sub-address family (SAFI), and all instances.
<afi> <safi> <prefix>	BGP routing table entry for the given prefix and all instances.
<afi> <safi> instance <instance-name>	BGP routing table summary for the given AFI, SAFI, and instance.
<afi> <safi> instance <instance-name> detail	Detailed list view of BGP routing table for the given AFI, SAFI, and instance.
<afi> <safi> instance <instance-name> <prefix>	BGP routing table entry for the given prefix and instance.
<afi> <safi> peer <name> / peer address <ip>	Peer name or address

Example 1: Summary view of the BGP rib-in.

```
supervisor@rtbrick: op> show bgp rib-in
Instance: ip2vrf, AFI: ipv4, SAFI: unicast
  Peer: None, Received routes: 10
    Prefix                              Next Hop                            MED        Local Preference  AS
Path
    198.51.100.75/24                    198.51.100.93                           -      100               -
    198.51.100.76/24                    198.51.100.94                       -          100               -
    198.51.100.77/24                    198.51.100.99                       -          100               -
    198.51.100.78/24                    198.51.100.94                       -          100               -
    198.51.100.79/24                    198.51.100.99                       -          100               -
    198.51.100.82/24                    198.51.100.94                       -          100               -
    198.51.100.93/24                    198.51.100.93                           -      100               -
    198.51.100.94/24                    198.51.100.94                       -          100               -
    198.51.100.99/24                    198.51.100.99                       -          100               -
    198.51.100.99/24                    198.51.100.99                       -          100              -
Instance: default, AFI: ipv4, SAFI: vpn-unicast
  Peer: None, Received routes: 4
    Prefix                              Next Hop                            MED        Local Preference  AS
Path
    198.51.100.14/24                    2001:db8:0:1::/32                   0          -
4200000004
    198.51.100.17/24                    2001:db8:0:1::/32                   0          -
4200000004
    198.51.100.16/24                    2001:db8:0:1::/32                   0          -
4200000004
```

Example 2: Summary view of the BGP rib-in for IPv4, all SAFIs and all instances

```
supervisor@rtbrick: op> show bgp rib-in ipv4
Instance: ip2vrf, AFI: ipv4, SAFI: unicast
  Peer: None, Received routes: 10
    Prefix                              Next Hop                            MED        Local Preference  AS
Path
    198.51.100.75/24                    198.51.100.93                       -          100              -
    198.51.100.76/24                    198.51.100.94                       -          100              -
    198.51.100.77/24                    198.51.100.95                       -          100              -
    198.51.100.78/24                    198.51.100.94                       -          100              -
    198.51.100.79/24                    198.51.100.95                       -          100              -
    198.51.100.82/24                    198.51.100.94                       -          100              -
    198.51.100.93/24                    198.51.100.93                       -          100              -
    198.51.100.94/24                    198.51.100.94                       -          100              -
    198.51.100.95/24                    198.51.100.95                       -          100              -
    198.51.100.99/24                    198.51.100.95                       -          100              -
Instance: default, AFI: ipv4, SAFI: vpn-unicast
  Peer: None, Received routes: 4
    Prefix                              Next Hop                            MED        Local Preference  AS
Path
    198.51.100.14/24                    2001:db8:0:13::                     0          -
4200000004
    198.51.100.17/24                    2001:db8:0:13::                     0          -
4200000004
```

Example 3: Summary view of the received routes

```
supervisor@rtbrick: op> show bgp rib-in ipv4 unicast peer address 198.51.100.94
Instance: ip2vrf, AFI: ipv4, SAFI: unicast
  Peer: None, Received routes: 13
    Prefix                              Next Hop                            MED        Local Preference  AS
Path
    198.51.100.75/24                    198.51.100.93                       -          100              -
    198.51.100.76/24                    198.51.100.94                       -          100              -
    198.51.100.77/24                    198.51.100.95                       -          100              -
    198.51.100.78/24                    198.51.100.94                       -          100              -
    198.51.100.79/24                    198.51.100.95                       -          100              -
    198.51.100.82/24                    198.51.100.94                       -          100              -
    198.51.100.93/24                    198.51.100.93                       -          100              -
    198.51.100.113/24                   198.51.100.93                       -          100
4200000003
    198.51.100.114/24                   198.51.100.93                       -          100
4200000004
    198.51.100.94/24                    198.51.100.94                       -          100              -
    198.51.100.95/24                    198.51.100.95                       -          100              -
    198.51.100.99/24                    198.51.100.95                       -          100              -
```

BGP RIB-out

This command displays the send routes.

Syntax:

show bgp rib-out <option> …

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

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

```
supervisor@rtbrick: op> show bgp rib-out ipv4 unicast peer CE1-Vrf1
Instance: vrf1, AFI: ipv4, SAFI: unicast
  Peer: CE1-Vrf1, Sent routes: 4
    Prefix                              MED             Local Preference  Origin          Next Hop
AS Path
    198.51.100.104/24                   0               -                 Incomplete      -
65001
    198.51.100.113/24                   0               -                 Incomplete      -
65001
    198.51.100.117/24                        0               -                 Incomplete      -
65001
    198.51.100.106/24                   0               -                 Incomplete      -
```

```
65001
```

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

```
supervisor@rtbrick: op> show bgp rib-out
Instance: vrf1, AFI: ipv4, SAFI: unicast
  Peer-group: pe1_ce1, Sent routes: 4
    Prefix                              MED             Local Preference  Origin
Next Hop                            AS Path
    198.51.100.104/24                   0               -
Incomplete      -                                  65001
    198.51.100.113/24                   0               -
Incomplete      -                                  65001
    198.51.100.105/24                   0               -
Incomplete      -                                  65001
    198.51.100.106/24                   0               -
Incomplete      -                                  65001
Instance: vrf1, AFI: ipv6, SAFI: unicast
  Peer-group: pe1_ce1, Sent routes: 3
    Prefix                              MED             Local Preference  Origin
Next Hop                            AS Path
    2001:db8:0:14::/24                  0               -
Incomplete      -                                  65001
    2001:db8:0:15::/24                  0               -
Incomplete      -                                  65001
    2001:db8:0:16::/24                  0               -
Incomplete      -                                  65001
```

TCP Connections

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

Syntax:

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

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

Example 1: Summary view of the BGP TCP connections

```
supervisor@leaf1: cfg> show bgp tcp bgp.iod.1 connection
Instance       Local IP Address                  Remote IP Address           Local
Port  Remote Port  State
default        2001:db8:0:189::                  2001:db8:0:38::             179
49568        Established
default        2001:db8:0:61::                   2001:db8:0:237::            50529
179          Established
```

Example 2: Detailed information of the BGP TCP connections

```
supervisor@leaf1: cfg> show bgp tcp bgp.iod.1 connection detail
Instance: default
  Local IPv6 address       : 2001:db8:0:189::
  Remote IPv6 address      : 2001:db8:0:38::
  Local port               : 179
  Remote port              : 49568
  State                    : Established
    Internal
      Options                  : -- | Keepalive | --
      TOS                      : 0
      TTL                      : 1
      Priority                 : 1
      Flags                    : -|-|-|-|-|Nagle Disabled|-|Wnd Scale|-|-|-
      Last trigger             : 27
      Timer                    : 37624
    Timers
      Poll                     : 0s
      Poll interval            : 0s
      Retransmission           : 65535s
    Receiver
      Expected sequence        : 32965979
      Available window         : 96816
      Announced window         : 95562
      Announced wnd RT edge    : 33061541
      MSS                      : 1440
      RTT estimate             : 0
    Timeout
      Sequence                 : 17683639s
      Retransmission           : 3s
      Retransmissions          : 0s
      Duplicate acks           : 0s
      Highest ack'd sequence   : 17683658s
    Congestion
      Window                   : 162834
      Persist count            : 0
      Send scale               : 5
      Receive scale            : 5
    Sender
      Next seq to send         : 17683658
      Last wnd update seq      : 32965979
      Last wnd update ack      : 17683658
      Window                   : 96192
      Max window announced     : 96800
      Acknowledged             : None
      Send buf                 : 56476
      Send queue length        : 0
      Unsent oversize          : 0
      TS last ack sent         : 818020352
    Keepalive
      Next keepalive idle      : 7200000
      Keepalive interval       : 75000
      Keepalive count          : 9
      Keep sent count          : 0
    Authentication
      Auth type                : HMAC-SHA-256-128
      key1-id                  : 255
      key2-id                  : 0
      Algorithm mismatch       : 0
      Secret mismatch          : 43
      Latest sent digest       : 850fea02c98912ce4497ec2b101a4f7c
      Latest received digest   : a718fb88e0d7fd4a00843e6aec03c864
```

TCP Statistics

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

Syntax:

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

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

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

```
supervisor@rtbrick: op> show bgp tcp bgp.iod.1 statistics instance default
Instance: default
  IP statistics
    Transmitted packets    : 3103242412
    Received packets       : 47351
    Forwarded packets      : 0
    Dropped packets        : 0
    Checksum error         : 0
    Invalid length error   : 0
    Out of memory error    : 0
    Routing error          : 0
    Protocol error         : 0
    Error in options       : 0
    Misc error             : 0
    Cachehit               : 0
  TCP statistics
    Transmitted packets    : 365499779
    Received packets       : 5577
```

```
     Forwarded packets      : 3014656
     Dropped packets        : 46
     Checksum error         : 0
     Invalid length error   : 0
     Out of memory error    : 0
     Routing error          : 3014656
     Protocol error         : 46
     Error in options       : 0
     Misc error             : 2097152
     Cachehit               : 1557594144
```

BGP Clear Commands

Clear commands allow to reset operational states.

BGP Peer

This commands resets BGP peerings.

Syntax:

clear bgp peer <option> …

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

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

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

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

2.3. IS-IS

2.3.1. IS-IS Overview

IS-IS, or Intermediate System to Intermediate System, is an open standard routing protocol. ISO published the standard as a way to route datagrams as part of their OSI stack. IETF later republished the standard, and added IP route support.

It is a link-state routing protocol, similar to OSPF. It forms neighbor adjacencies, has areas, exchanges link-state packets, builds a link-state database and runs the Dijkstra SPF algorithm to find the best path to each destination, which is installed in the routing table.

Segment Routing

IS-IS in RBFS supports segment routing based on RFC 8667. IS-IS segment routing extensions allow to advertise labels with prefixes.

> RFC and draft compliance are partial except as specified.

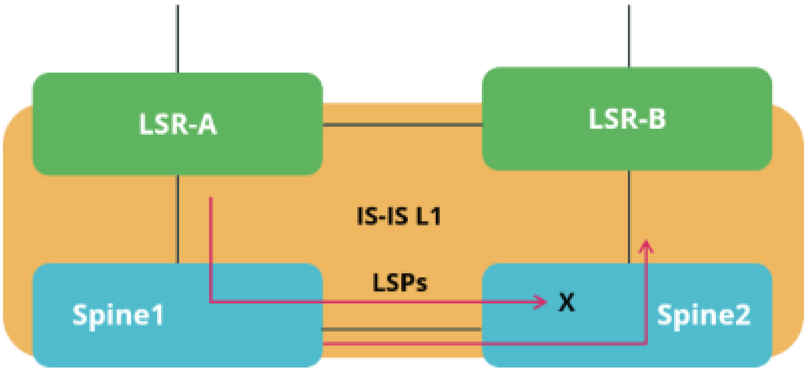
RBFS currently supports the following IS-IS segment routing features:

- MPLS data plane
- IPv4 prefixes (TLV 135) and IPv6 prefixes (TLV 236)
- Prefix SID with node flag (Node SID) on loopback interface
- Anycast SID
- A single global SRGB block
- Adjacency SIDs

IS-IS Flood Filter Configuration

In IS-IS, by default all routers flood link-state packets, so that all routers will have a complete topology view. IS-IS flood filters allow to modify this behavior and limit the exchange of LSPs. For example, if two spine routers in a spine/leaf fabric are symmetrically connected to two upstream label-switch routers (LSR) like shown in the figure below, you can use a flood filter to not advertise LSPs learned from LSR A back to the LSR B via the second spine switch.

The flooding filter configuration is part of the global configuration hierarchy and therefore you can configure filtering globally, i.e. not per instance, so that the filter configurations can be reused across instances.



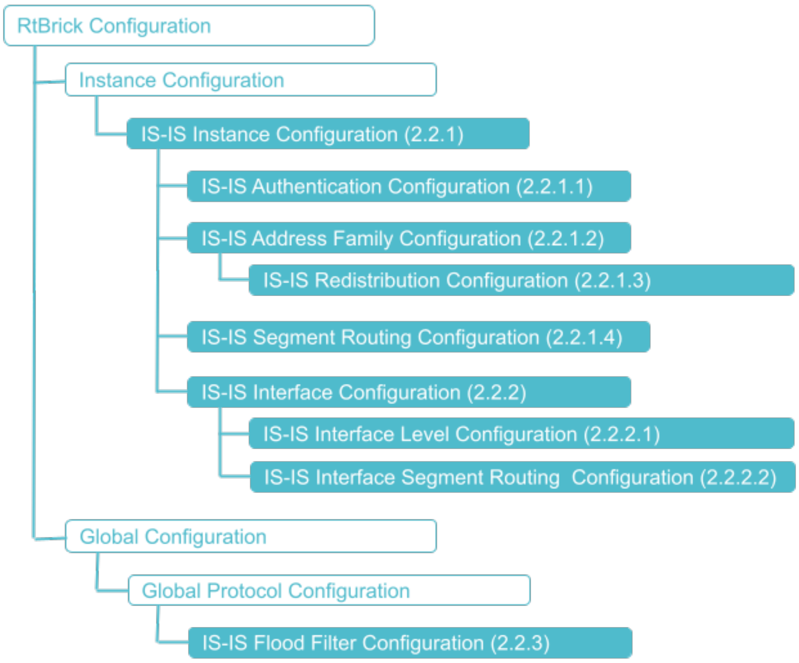
Supported Platforms

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

2.3.2. IS-IS Configuration

Configuration Hierarchy

The diagram below illustrates the IS-IS configuration hierarchy.


RtBrick Configuration
Instance Configuration
IS-IS Instance Configuration (2.2.1)
IS-IS Authentication Configuration (2.2.1.1)
IS-IS Address Family Configuration (2.2.1.2)
IS-IS Redistribution Configuration (2.2.1.3)
IS-IS Segment Routing Configuration (2.2.1.4)
IS-IS Interface Configuration (2.2.2)
IS-IS Interface Level Configuration (2.2.2.1)
IS-IS Interface Segment Routing Configuration (2.2.2.2)
Global Configuration
Global Protocol Configuration
IS-IS Flood Filter Configuration (2.2.3)

Configuration Syntax and Commands

The following sections describe the IS-IS configuration syntax and commands.

Instance Configuration

The instance configuration hierarchy includes parameters that are required for or used by IS-IS.

Syntax:

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

Attribute	Description
<name>	Name of the IS-IS instance
area <area>	IS-IS area-address. The area can be represented in 1, 3, 5, 13 bytes format.
authentication <…>	Specifies the authentication scheme for IS-IS. Refer to section 2.2.1.1 for the IS-IS authentication configuration details.
holding-time <holding-time>	Specifies how long a neighbor should consider this routing device to be operative without receiving another hello packet. Default value: 30 seconds Range: 1 through 3,180 seconds
hostname <hostname>	Specifies the hostname mapped to the system identifier.
ignore-attached-bit [true/false]	This configuration allows you to enable the routing device to ignore the attached bit on incoming Level 1 link-state PDUs. If the attached bit is ignored, no default route, which points to the routing device which has set the attached bit, is installed.
interface <…>	Name of the interface. Refer to section 2.2.2 for the interface configuration details.
ipv6-disable [true/false]	Specifies whether the ipv6-disable configuration is enabled or not. When you set this value to "true", it indicates that IPv6 configuration is disabled.
[level-1/level-2] address-family <…>	Protocol ISIS level-1/level-2 address-family configuration. Refer to section 2.2.1.2 for the address family configuration details.
level1-to-level2 route-leak [enable/disable]	Specifies whether the level1-to-level2 route-leak is enabled or not. When set to disable, IS-IS will not leak routing information from a Level 1 area to a Level 2 area. By default, this option is enabled.
lsp-lifetime <lsp-lifetime>	IS-IS link-state PDUs maximum lifetime, default 65535 seconds

Attribute	Description
multipath <multipath>	Load sharing among multiple ISIS paths, default 256
no-mpls-transit-path [true/false]	When set to true, IS-IS will not install segment routing transit path, default false
overload [true/false]	When set to true, IS-IS overload bit is set, default false
router-id <router-id>	ISIS router identifier (ipv4 format: A.B.C.D)
system-id <system-id>	Specifies the system ID of the device.

Example 1: IS-IS Instance Configuration

```
{
    "rtbrick-config:instance": [
      {
        "name": "default",
        "protocol": {
          "isis": {
            "system-id": "1000.9900.0001",
            "overload": "false",
            "holding-time": 100,
            "area": "49.0001/24",
            "hostname": "spine1",
            "router-id": "198.51.100.199",
            "authentication": {
              "level-1": {
                "check": "enable"
              }
            }
          }
        }
      }
    ]
  }
```

Example 2: Disabling IS-IS Route Leaking from a Level 1 Area to a Level 2 Area

```
supervisor@rtbrick>spine1: cfg> show config instance default protocol isis level1-to-level2
{
  "rtbrick-config:level1-to-level2": {
    "route-leak": "disable"
  }
}
```

IS-IS Authentication Configuration

Syntax:

set instance <instance-name> **protocol isis authentication** [**level-1** | **level-2**] <attribute> <value>

Attribute	Description
check [disable / enable]	Specifies an authentication check to reject PDUs that do not match the type or key requirements. You can enable or disable the authentication check.
key-id1 <key-id1> / key-id2 <key-id2>	The key ID allows you to specify the key identifiers for level-1/level-2 authentication.
key1-encrypted-text <key1-encrypted-text> / key2-encrypted-text <key2-encrypted-text>	Authentication key1 and key 2 encrypted text
key1-plain-text <key1-plain-text> / key2-plain-text <key2-plain-text>	The level-1/level-2 authentication keys specify the authentication keys (passwords) that are used by the neighboring routing devices to verify the authenticity of packets sent from this interface. For the key to work, you also must include the authentication-type statement.
type	Enables you to specify the authentication scheme for IS-IS. If you enable authentication, you must specify a password by including the authentication-key statement. The following authentication types are supported: • clear_text • md5 • sha1

```
{
  "ietf-restconf:data": {
    "rtbrick-config:instance": [
      {
        "name": "default",
        "protocol": {
          "isis": {
            <...>
            "authentication": {
              "level-1": {
                "type": "md5",
                "key1-encrypted-text":
"$239928e897b1f0fb3a97ed426db21aba36ca48479744a7c71255ee2c4e747e859"
              }
            },
            "interface": [
              {
                "name": "ifl-0/0/1/0",
                "type": "point-to-point",
                "level-1": {
                  "snp-authentication": "enable",
                  "hello-authentication": "disable",
                  "metric": 10
                },
                "level-2": {
                  "snp-authentication": "enable",
                  "hello-authentication": "disable",
                  "metric": 10
                }
              },
              <...>
```

IS-IS Address-Family Configuration

The address-family command allows you to enable the address families that IS-IS will route and configure settings that are specific to that address family.

Syntax:

set instance <instance-name> **protocol isis** [**level-1** | **level-2**] **address-family** <attribute> <value>

Attribute	Description
<afi>	Address family identifier (AFI). Supported values: ipv4, ipv6
<safi>	Subsequent address family identifier (SAFI). Supported values: unicast or labeled-unicast

Configuring Route Redistribution

Syntax:

set instance <instance-name> **protocol isis** [**level-1** | **level-2**] **address-family** <afi> <safi> **redistribute** <attribute> <value>

Attribute	Description
<afi>	Address family identifier (AFI). Supported values: ipv4, ipv6
<safi>	Subsequent address family identifier (SAFI). Supported values: unicast or labeled-unicast
redistribute <protocol>	Specifies the source from which the routes are to be redistributed from. The available options include arp-nd, bgp, bgp-local, bgp-local-origin, direct, igmp, ospf, l2tpv2, ldp, local, pim, ppp, rib, and static.
redistribute <protocol> <policy>	Specifies the name of the policy map. The redistribute attach point allows routes from other sources to be advertised by IS-IS. Policy can be applied only to the routes that are redistributed from other sources to IS-IS. The support for inter-level leaking through policy is unavailable.

Example: IS-IS address-family configuration

```
{
    "rtbrick-config:isis": {
      "system-id": "1000.9900.0001",
      "area": "49.0001/24",
      "hostname": "spine1",
      "interface": [
        {
          "name": "ifl-0/0/2/0",
          "type": "point-to-point",
          "level-1": {
            "metric": 1000
          }
        }
      ],
      "level-1": {
        "address-family": [
          {
            "afi": "ipv4",
            "safi": "unicast",
            "redistribute": [
              {
                "source": "bgp"
                "policy": "filter-link-address"

              }
            ]
          }
        ]
      }
    }
```

Segment Routing Configuration

Syntax:

set instance <instance-name> **protocol isis segment-routing** <attribute> <value>

Attribute	**Description**
srgb base <srgb base>	Specifies the segment routing global block (SRGB) in source packet routing. SRGB is used for prefix SIDs. Supported MPLS label values are 0 - 1048575. The reserved MPLS label range is 0 - 15. In RBFS, BGP uses the label range 20000 - 100000. It is recommended to assign label values outside of these reserved ranges to avoid conflicts.
srgb range <srgb range>	IS-IS system range of labels from the base label.
srlb base <srlb base>	Specifies the segment routing local block (SRLB) in source packet routing. SRLB is used for adjacency SIDs. Supported MPLS label values are 0 - 1048575. The reserved MPLS label range is 0 - 15. In RBFS, BGP uses the label range 20000 - 100000. It is recommended to assign label values outside of these reserved ranges to avoid conflicts.
srlb range <srlb range>	IS-IS system range of labels from the base label.

Example: IS-IS Segment Routing Configuration

```
{
    "rtbrick-config:isis": {
      segment-routing: {
        "srgb: {
          "base": 5000,
          "range": 1000
        }
        "srlb: {
          "base": 5000,
          "range": 1000,
        }
```

Configuring IS-IS Interface

By default, there are no interfaces associated with IS-IS. You must configure at least one IS-IS interface for IS-IS adjacency formation.

Syntax:

set instance <instance> **protocol isis interface** <name> <attribute> <value>

Attribute	Description
<name>	Specifies the name of the IS-IS interface.
flood-filter <flood-filter>	Specifies the IS-IS flood filter name
level-1 / level-2	Specify IS-IS interface level configuration. Refer to section 2.2.2.1 for the IS-IS interface level configuration details.
lsp-interval <lsp-interval>	IS-IS system interface LSP interval, default 100
passive [true / false]	Enable interface in passive mode, default false
system-id <system-id>	Interface level system id
type [loopback / none / point-to-point]	Specifies the type of the IS-IS system interface
ldp-synchronization [enable / disable]	Enable LDP IGP synchronization, default disable

Example 1: IS-IS Interface Configuration

```
{
    "rtbrick-config:isis": {
      "interface": [
        {
          "name": "ifl-0/0/1/0",
          "lsp-interval": 200
        }
      ]
    }
  }
```

Example 2: IS-IS Interface Level Flood Filter Configuration

```
{
    "rtbrick-config:interface": [
      {
        "name": "ifl-0/0/1/0",
        "flood-filter": "spine1_lsr1_flood_filter"
      }
    ]
  }
```

Example 3: IS-IS Interface Configuration with enabled LDP synchronization

```
{
    "rtbrick-config:isis": {
      "interface": [
        {
          "name": "ifl-0/0/1/0",
          "lsp-interval": 200,
          "ldp-synchronization": "enable"
        }
      ]
    }
  }
```

IS-IS Interface Level Configuration

Syntax:

set instance <instance> **protocol isis interface** <name> [**level-1** | **level-2**] <attribute> <value>

Attribute	Description
adjacency-disable [true/false]	Specify the level-1/level-2 adjacency on an interface, default false
hello-authentication [disable/enable]	Authentication on hello packets
metric <metric>	Level-1/Level-2 metric on an interface, default 1000000
snp-authentication [enable/disable]	Authentication on CSNP/PSNP packets

Example: IS-IS Interface Level Configuration

```
{
    "rtbrick-config:interface": [
      {
        "name": "ifl-0/0/1/0",
        "lsp-interval": 200,
        "level-1": {
          "snp-authentication": "enable",
          "hello-authentication": "enable",
          "metric": 1000,
          "adjacency-disable": "false"
        }
      }
    ]
  }
```

Interface-level Segment Routing Configuration

Syntax:

set instance <instance> **protocol isis interface** <name> **segment-routing** <attribute> <value>

Attribute	Description
segment-routing [ipv4 / ipv6] anycast-index <anycast-index>	Anycast index segment-ID. The prefix SIDs and anycast SIDs are applied on loopback interface only.
segment-routing [ipv4 / ipv6] index <index>	Prefix index segment ID.
segment-routing point-to-point [ipv4 / ipv6] adjacency-index <adjacency-index>	Adjacency index segment-ID. The adjacency SIDs are applied on active IS-IS interfaces on which adjacencies are established.

Example 1: IS-IS Interface Level Segment Routing Configuration for Prefix and Anycast SID

```
"rtbrick-config:instance": [
     {
       "name": "default",
       "protocol": {
         "isis": {
           "interface": [
             {
               "name": "lo-0/0/0",
               "segment-routing": {
                 "ipv4": {
                   "index": 100
                 },
                 "ipv6": {
                   "index": 200
                 }
               }
             },
             {
                "name": "lo-0/0/1",
                "segment-routing": {
                  "ipv4": {
                    "anycast-index": 110
                  },
                  "ipv6": {
                    "anycast-index": 210
                  }
                }
              }
            ]
          }
        }
      }
    ]
  }
}
```

Example 2: IS-IS Interface Level Segment Routing Configuration for Adjacency SID

```
{
  "rtbrick-config:isis": {
    "interface": [
      {
        "name": "ifp-0/0/1/0",
        "type": "point-to-point",
        "segment-routing": {
          "point-to-point": {
            "ipv4": {
              "adjacency-index": 241
            },
            "ipv6": {
              "adjacency-index": 261
            }
          }
        }
      }
    ]
  }
}
```

IS-IS Global Configuration

IS-IS Flood Filter Configuration

Syntax:

set global protocol isis flood-filter <filter-name> <ordinal> <attribute> <value>

Attribute	Description
<filter-name>	Filter-name which binds a flooding filter to an IS-IS interface
<ordinal>	Number to filter rule
action [block/flood]	Action required to flood or not
ordinal-name <ordinal-name>	Name for the filter rule
system-id <system-id>	IS-IS instance system-id
system-id-mask <system-id-mask>	System ID mask on which the filter should match

Example: IS-IS Flood Filter Configuration

```
{
    "rtbrick-config:flood-filter": [
      {
        "filter-name": "spine1_lsr1_flood_filter",
        "ordinal": 1,
        "ordinal-name": "spine1",
        "system-id": "1920.0100.4001",
        "action": "flood"
      }
    ]
  }
```

2.3.3. IS-IS Operational Commands

IS-IS Show Commands

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

IS-IS Overview

Syntax:

show isis overview

Option	Description
-	Without any option, this command displays a summary of all the IS-IS instances

Example: Summary view of all the IS-IS instances

```
supervisor@rtbrick>spine1: op> show isis overview
Instance: default
  System ID: 1921.6800.1002
  System hostname: No hostname configured
  Areas: 49.0001/24
  Neighbor hold time: 30 sec
  LSP life time: 65535 se
  Overload bit set: False
  SRGB base: not defined
  SRGB range: not defined
  SRGB label values: not defined
  SRLB base: not defined
  SRLB range: not defined
  SRLB label values: not defined
  Authentication: Level 1: none, Level 2: none
```

IS-IS Interface

Syntax:

show isis interface <option>

Option	Description
-	Without any option, this command displays a summary of all the IS-IS interfaces
instance	Displays IS-IS interface information for an instance
statistics	Displays IS-IS interface statistics information
detail	Displays detailed output for all interfaces.

Example 1: Summary view of the IS-IS interfaces

```
supervisor@rtbrick>spine1: op> show isis interface
Instance: default
  Interface         Level   Adjacencies  Metric     Type                  Passive
  lo-0/0/4/1          1             0   1000000     loopback                True
  ifl-0/1/2/12        1             0   1000000     point-to-point          False
  ifl-0/1/6/16        1             0   1000000     point-to-point          False
```

Example 2: Summary view of the IS-IS interfaces for a specific instance

```
supervisor@rtbrick>spine1: op> show isis interface instance default
Instance: default
  Interface         Level   Adjacencies   Metric    Type                Passive
  lo-0/0/4/1            1           0     1000000   loopback            True
  ifl-0/1/2/12          1           0     1000000   point-to-point      False
  ifl-0/1/6/16          1           0     1000000   point-to-point      False
```

Example 3: Summary view of the IS-IS interfaces for a specific interface

```
supervisor@rtbrick>spine1: op> show isis interface ifl-0/1/6/16
Instance: default
  Interface: ifl-0/1/6/16, Level: 1
    Type: point-to-point, Passive: False
    Metric: 1000000
    Adjacencies: 1
    CNSP: In: 24 Out: 34 Success: 24 Fail: 0
    PSNP: In: 8 Out: 10 Success: 5 Fail: 1
    LSP:  In: 14 Out: 11 Success: 11 Fail: 2 In Purge: 0 In Auth Fail: 2
    IIH:  In: 121 Out: 163
```

Example 4: Summary view of the IS-IS interface statistics

```
supervisor@rtbrick>spine1: op> show isis interface statistics
Instance: default
  Interface     Level  CSNP In  CSNP Out  CSNP Fail  PSNP In  PSNP Out  PSNP Fail
LSP In  LSP Out  LSP Fail  IIH In  IIH Out
  lo-0/0/4/1        1       0       0       0       0       0       0       0
0       0       0       0
  ifl-0/1/2/12      1       32      32          0       9       6       1       9
10      0   117     138
  ifl-0/1/6/16      1       22      32          0       6       6       1       9
8       0   115     138
```

Example 5: Detailed output of the IS-IS interface

```
supervisor@rtbrick>spine1: op> show isis interface detail
Instance: default
  Interface: lo-0/0/0/0, Level: 1
    Type: loopback, Passive: True
    Metric: 1000000
    Adjacencies: 0
    CNSP: In: 0 Out: 0 Success: 0 Fail: 0
    PSNP: In: 0 Out: 0 Success: 0 Fail: 0
    LSP:  In: 0 Out: 0 Success: 0 Fail: 0 In Purge: 0 In Auth Fail: 0
    IIH:  In: 0 Out: 0
Instance: default
  Interface: ifl-0/0/0/0, Level: 1
    Type: point-to-point, Passive: False
    Metric: 1000000
    Adjacencies: 1
    CNSP: In: 3020 Out: 3020 Success: 3020 Fail: 0
    PSNP: In: 2 Out: 2 Success: 2 Fail: 0
    LSP:  In: 2 Out: 2 Success: 2 Fail: 0 In Purge: 0 In Auth Fail: 0
    IIH:  In: 5589 Out: 5600
```

IS-IS Neighbor

Syntax:

show isis neighbor <option>

Option	Description
-	Without any option, this command displays a summary of all the IS-IS neighbors
detail	Displays detailed information for IS-IS neighbor
instance	Displays IS-IS neighbor information for an instance

Example 1: Summary view of the IS-IS neighbor

```
supervisor@rtbrick>spine1: op> show isis neighbor
Instance: default
  Interface      System          Level   State   Type    Up since
Expires
  ifl-0/1/2/12  1920.0100.4002.00  L1    Up      P2P     Mon Nov 02 06:18:36     in
28s 228094us
  ifl-0/1/6/16  1920.0000.0006.00  L1    Up      P2P     Mon Nov 02 06:18:30     in
24s 420225us
```

Example 2: Detailed view of the IS-IS neighbor

```
supervisor@S1-STD-17-1703: cfg> show isis neighbor detail
Instance: ip2vrf
  System: isr6, Interface: ifl-0/0/2/0
    State: Up, Level: L1, Adjacency type: P2P
    Holding time: 30.0s, Expiry time: in 25s 332949us
    Local IPv4 address: 198.51.100.27, Remote IPv4 address: 198.51.100.28
    Local IPv6 address: 2001:db8:0:76::, Remote IPv6 address: 2001:db8:0:34::
    IPv4 Adjacency SID: 11116, IPv6 Adjacency SID: 11117
    Up since: Wed Feb 16 04:46:25 GMT +0000 2022, Last down reason: Admin reset
    Last transition: 2022-02-16T04:46:25.300144+0000, Number of transitions: 14
    Error counters:
      Level mismatch: 0, Area mismatch: 0, System ID: 0, Subnet mismatch: 0
      Hold timeout: 3, Neighbor down: 0, Interface down: 0, Admin reset: 1
      Interface configuration: 0, Area configuration: 0, Other: 0
```

Example 3: Summary view of the IS-IS neighbor for the specified instance

```
supervisor@rtbrick>spine1: op> show isis neighbor instance default
Instance: default
  Interface         System          Level   State   Type    Up since
Expires
  ifl-0/1/2/12  1920.0100.4002.00  L1   Up      P2P     Mon Nov 02 06:18:36     in
28s 678329us
  ifl-0/1/6/16  1920.0000.0006.00  L1   Up      P2P     Mon Nov 02 06:18:30     in
28s 88085us
supervisor@rtbrick>spine1: op>
```

Example 4: Detailed view of the IS-IS neighbor for the specified instance

```
supervisor@rtbrick>spine1: op> show isis neighbor instance default detail
Instance: default
  System: 1920.0100.4002.00, Interface: ifl-0/1/2/12
    State: Up, Level: L1, Adjacency type: P2P
    Holding time: 30.0s, Expiry time: in 21s 706586us
    Local IPv4 address: 198.51.100.22, Remote IPv4 address: 198.51.100.21
    Local IPv6 address: 2001:db8:0:110::, Remote IPv6 address: 2001:db8:0:3433::
    Up since: Mon Nov 02 06:18:36 GMT +0000 2020, Last down reason: NA
    Last transition: 2020-11-02T06:18:36.947601+0000, Number of transitions: 2
    Error counters:
    Level mismatch: 0, Area mismatch: 0, System ID: 0, Subnet mismatch: 0
    Hold timeout: 0, Neighbor down: 0, Interface down: 0, Admin reset: 0
    Interface configuration: 0, Area configuration: 0, Other: 0
  System: 1920.0000.0006.00, Interface: ifl-0/1/6/16
    State: Up, Level: L1, Adjacency type: P2P
    Holding time: 30.0s, Expiry time: in 22s 832756us
    Local IPv4 address: 198.51.100.100, Remote IPv4 address: 198.51.100.101
    Local IPv6 address: 2001:db8:0:10::, Remote IPv6 address: 2001:db8:0:6843::
    Up since: Mon Nov 02 06:18:30 GMT +0000 2020, Last down reason: NA
    Last transition: 2020-11-02T06:18:30.356111+0000, Number of transitions: 2
    Error counters:
    Level mismatch: 0, Area mismatch: 0, System ID: 0, Subnet mismatch: 0
    Hold timeout: 0, Neighbor down: 0, Interface down: 0, Admin reset: 0
    Interface configuration: 0, Area configuration: 0, Other: 0
```

IS-IS Hostname

Syntax:

show isis hostname

Option	Description
-	Without any option, this command displays a summary of all the IS-IS dynamic hostnames

Example: Summary view of IS-IS hostnames

```
supervisor@rtbrick>spine1: op> show isis hostname
```

```
Instance    System-ID           Hostname
default         1920.0100.4001      spine1
supervisor@rtbrick>spine1: op>
```

IS-IS Database

Syntax:

show isis database <option>

Option	Description
-	Without any option, this command displays all the IS-IS databases
detail	Displays detailed information for IS-IS database
instance	Displays IS-IS database information for an instance
lsp <lsp-id>	Displays a summary of IS-IS database for the specified LSP ID. This command includes an option for entering the system ID part either by hostname or by ID.
[level-1/level-2] lsp	Displays a summary of IS-IS database LSP information for specified level
system <system-id>	Displays a summary of IS-IS database for all LSPs from a system
[level-1/level-2] system	Displays a summary of IS-IS database for all LSPs from a system on the specified level.
[level-1/level-2] detail	Displays detailed information for the specified level

Example 1: Summary view of the IS-IS Database

```
supervisor@S1-STD-7-7001>bm01-tst.fsn.rtbrick.net: op> show isis database
Instance: default, Level: 1
  LSP ID                            Sequence     Checksum     Lifetime     Overload     Attached
  1921.6800.1002.00-00           0x3          0x9561       65535            0            0
  1921.6800.1005.00-00           0x2          0x499b       65535            0            0
Instance: default, Level: 2
  LSP ID                             Sequence     Checksum     Lifetime     Overload     Attached
  1921.6800.1002.00-00           0x4           0x531a      65535              0          0
```

Example 2: Summary view of the IS-IS database for the specified LSP

```
supervisor@rtbrick>spine1: op> show isis database lsp 1920.0100.4001.00-00
Instance: default, Level: 1
```

```
  LSP ID: 1920.0100.4001.00-00
    Interface:
    LSP Header:
    Sequence: 0xc
    Checksum: 0x9c74
    Remaining lifetime: 65535 seconds
    Flags: Attached: 0, Overload: 0
    Packet:
    Length: 168 bytes
    Last received time: 2020-11-02T06:46:46.473726+0000
    Expiry: expires in 17h 52m 34s 950743us
    Dynamic Hostname TLV: spine1
    Protocols Supported TLVs:
    Network layer protocol ID: IPv6
    Network layer protocol ID: IPv4
    Area Address TLVs:
    Area address: 49.0001
    Authentication TLV:
    Value: 77b259cb36930819b0abb6120ceee2fd
    IS Reachability TLVs:
    IS neighbor: 1920.0000.0006.00
    IS neighbor: 1920.0100.4002.00
    IPv4 Reachability TLVs:
    IPv4 prefix: 198.51.100.100/24                  Metric:  1000000   Internal
Up
    IPv4 prefix: 198.51.100.22/24                   Metric:  1000000   Internal
Up
    IPv4 prefix: 198.51.100.41/24                   Metric:  1000000   Internal
Up    SID:    1   Flags: Node
    IPv6 Reachability TLVs:
    IPv6 prefix: 2001:db8:0:41::/32                 Metric:  1000000   Internal
Up
   Segment Routing TLVs:
    SRGB: Base: 10000, Range: 2000
Instance: default, Level: 2
  LSP ID: 1920.0100.4001.00-00
    Interface:
    LSP Header:
    Sequence: 0x12
    Checksum: 0x6407
    Remaining lifetime: 65535 seconds
    Flags: Attached: 0, Overload: 0
    Packet:
    Length: 247 bytes
    Last received time: 2020-11-02T06:47:06.466723+0000
    Expiry: expires in 17h 52m 54s 889789us
    Dynamic Hostname TLV: spine1
    Protocols Supported TLVs:
    Network layer protocol ID: IPv6
    Network layer protocol ID: IPv4
    Area Address TLVs:
    Area address: 49.0001
    Authentication TLV:
    none
    IS Reachability TLVs:
    IPv4 Reachability TLVs:
    IPv4 prefix: 198.51.100.100/24                  Metric:  1000000   Internal
Up
    IPv4 prefix: 198.51.100.30/24                   Metric:  2000000   Internal
Up
    IPv4 prefix: 198.51.100.20/24                   Metric:  2000000   Internal
```

```
Up
    IPv4 prefix: 198.51.100.21/24                   Metric:  2000000   Internal
Up
    IPv4 prefix: 198.51.100.22/24                   Metric:  1000000   Internal
Up
    IPv4 prefix: 198.51.100.26/24                   Metric:  2000000   Internal
Up    SID:    6   Flags: Re-advertisement, Node
    IPv4 prefix: 198.51.100.41/24                   Metric:  1000000   Internal
Up    SID:    1   Flags: Re-advertisement, Node
    IPv4 prefix: 198.51.100.42/24                   Metric:  2000000   Internal
Up    SID:    2   Flags: Re-advertisement, Node
    IPv6 Reachability TLVs:
    IPv6 prefix: 2001:db8:0:41::/32                 Metric:  1000000   Internal
Up
    IPv6 prefix: 2001:db8:0:42::/32                 Metric:  2000000   Internal
Up
   Segment Routing TLVs:
    SRGB: Base: 10000, Range: 2000
```

Example 3: Detailed view of the IS-IS database for level-1

```
supervisor@rtbrick>spine1: op> show isis database level-1 detail
Instance: default, Level: 1
  LSP ID: 1920.0100.4001.00-00
    Interface:
    LSP Header:
    Sequence: 0xc
    Checksum: 0x9c74
    Remaining lifetime: 65535 seconds
    Flags: Attached: 0, Overload: 0
    Packet:
    Length: 168 bytes
    Last received time: 2020-11-02T06:46:46.473726+0000
    Expiry: expires in 17h 50m 31s 759013us
    Dynamic Hostname TLV: spine1
    Protocols Supported TLVs:
    Network layer protocol ID: IPv6
    Network layer protocol ID: IPv4
    Area Address TLVs:
    Area address: 49.0001
    Authentication TLV:
    Value: 77b259cb36930819b0abb6120ceee2fd
    IS Reachability TLVs:
    IS neighbor: 1920.0000.0006.00
    IS neighbor: 1920.0100.4002.00
    IPv4 Reachability TLVs:
    IPv4 prefix: 198.51.100.100/24                  Metric:  1000000   Internal
Up
    IPv4 prefix: 198.51.100.22/24                   Metric:  1000000   Internal
Up
    IPv4 prefix: 198.51.100.41/24                   Metric:  1000000   Internal
Up    SID:    1   Flags: Node
    IPv6 Reachability TLVs:
    IPv6 prefix: 2001:db8:0:41::/32                 Metric:  1000000   Internal
Up
   Segment Routing TLVs:
    SRGB: Base: 10000, Range: 2000
  LSP ID: 1920.0100.4002.00-00
    Interface: ifl-0/1/2/12
    LSP Header:
    Sequence: 0x9
    Checksum: 0x89a6
    Remaining lifetime: 65534 seconds
    Flags: Attached: 0, Overload: 0
    Packet:
    Length: 149 bytes
    Last received time: 2020-11-02T06:45:59.814186+0000
    Expiry: expires in 17h 49m 44s 99010us
    Dynamic Hostname TLV: none
    Protocols Supported TLVs:
    Network layer protocol ID: IPv6
    Network layer protocol ID: IPv4
    Area Address TLVs:
    Area address: 49.0001
    Authentication TLV:
    Value: 5892f2d37d7f23abcfcb48466276659c
    IS Reachability TLVs:
    IS neighbor: 1920.0100.4001.00
    IPv4 Reachability TLVs:
    IPv4 prefix: 198.51.100.30/24               Metric:  1000000   Internal   Up
    IPv4 prefix: 198.51.100.22/24               Metric:  1000000   Internal   Up
    IPv4 prefix: 198.51.100.42/24               Metric:  1000000   Internal   Up
SID:    2   Flags: Node
    IPv6 Reachability TLVs:
    IPv6 prefix: 2001:db8:0:42::/32             Metric:  1000000   Internal   Up
   Segment Routing TLVs:
    SRGB: Base: 70000, Range: 2000
```

Example 4: Summary view of the IS-IS database for the specified instance

```
supervisor@rtbrick>spine1: op> show isis database instance default
Instance: default, Level: 1
  LSP ID                    Sequence    Checksum    Lifetime    Expiry                  Overload    Attached
  1920.0000.0006.00-00          0xa     0x8beb      65534   in 17h 44m 13s 390502us        0           0
  1920.0000.0007.00-00          0x5     0xdfbb      65533   in 17h 15m 49s 869683us        0           0
  1920.0000.0008.00-00          0x6     0x76f3      65533   in 17h 15m 51s 959359us        0           0
  1920.0000.0009.00-00          0x6     0x5b18      65533   in 17h 15m 51s 667570us        0           0
  spine1.00-00                  0xc     0x9c74      65535   in 17h 43m 55s 63659us         0           0
  1920.0100.4002.00-00          0x9     0x89a6      65534   in 17h 43m 7s 403686us         0           0
Instance: default, Level: 2
  LSP ID                    Sequence    Checksum    Lifetime    Expiry                  Overload    Attached
  spine1.00-00                  0x12    0x6407      65535   in 17h 44m 15s 21189us         0           0
```

Example 5: Summary view of the IS-IS database for the specified instance and LSP

```
supervisor@rtbrick>spine1: op> show isis database instance default lsp
1920.0100.4001.00-00
Instance: default, Level: 1
  LSP ID: 1920.0100.4001.00-00
    Interface:
    LSP Header:
    Sequence: 0xc
    Checksum: 0x9c74
    Remaining lifetime: 65535 seconds
    Flags: Attached: 0, Overload: 0
    Packet:
    Length: 168 bytes
    Last received time: 2020-11-02T06:46:46.473726+0000
    Expiry: expires in 17h 52m 34s 950743us
    Dynamic Hostname TLV: spine1
    Protocols Supported TLVs:
    Network layer protocol ID: IPv6
    Network layer protocol ID: IPv4
    Area Address TLVs:
    Area address: 49.0001
    Authentication TLV:
    Value: 77b259cb36930819b0abb6120ceee2fd
    IS Reachability TLVs:
    IS neighbor: 1920.0000.0006.00
    IS neighbor: 1920.0100.4002.00
    IPv4 Reachability TLVs:
    IPv4 prefix: 198.51.100.100/24              Metric:  1000000   Internal   Up
    IPv4 prefix: 198.51.100.22/24               Metric:  1000000   Internal   Up
    IPv4 prefix: 198.51.100.41/24               Metric:  1000000   Internal   Up
SID:    1   Flags: Node
    IPv6 Reachability TLVs:
    IPv6 prefix: 2001:db8:0:41::/32             Metric:  1000000   Internal   Up
   Segment Routing TLVs:
    SRGB: Base: 10000, Range: 2000
Instance: default, Level: 2
  LSP ID: 1920.0100.4001.00-00
    Interface:
    LSP Header:
    Sequence: 0x12
    Checksum: 0x6407
    Remaining lifetime: 65535 seconds
    Flags: Attached: 0, Overload: 0
    Packet:
    Length: 247 bytes
    Last received time: 2020-11-02T06:47:06.466723+0000
    Expiry: expires in 17h 52m 54s 889789us
    Dynamic Hostname TLV: spine1
    Protocols Supported TLVs:
    Network layer protocol ID: IPv6
    Network layer protocol ID: IPv4
    Area Address TLVs:
    Area address: 49.0001
    Authentication TLV:
    none
    IS Reachability TLVs:
    IPv4 Reachability TLVs:
    IPv4 prefix: 198.51.100.100/24              Metric:  1000000   Internal   Up
    IPv4 prefix: 198.51.100.30/24               Metric:  2000000   Internal   Up
    IPv4 prefix: 198.51.100.20/24               Metric:  2000000   Internal   Up
    IPv4 prefix: 198.51.100.21/24               Metric:  2000000   Internal   Up
    IPv4 prefix: 198.51.100.22/24               Metric:  1000000   Internal   Up
    IPv4 prefix: 198.51.100.26/24               Metric:  2000000   Internal   Up
SID:    6   Flags: Re-advertisement, Node
    IPv4 prefix: 198.51.100.41/24               Metric:  1000000   Internal   Up
SID:    1   Flags: Re-advertisement, Node
    IPv4 prefix: 198.51.100.42/24               Metric:  2000000   Internal   Up
SID:    2   Flags: Re-advertisement, Node
    IPv6 Reachability TLVs:
    IPv6 prefix: 2001:db8:0:41::/32             Metric:  1000000   Internal   Up
    IPv6 prefix: 2001:db8:0:42::/32             Metric:  2000000   Internal   Up
   Segment Routing TLVs:
    SRGB: Base: 10000, Range: 2000
```

IS-IS Route

Syntax:

show isis route <option>

Option	Description
-	Without any option, this command displays a summary of the IS-IS route information
instance	Displays IS-IS route information for an instance
<afi> <safi>	Routing information for the specified AFI/SAFI. Supported SAFI values are 'unicast' and 'labeled-unicast'.

Example 1: Summary view of the IS-IS routes

```
supervisor@rtbrick>spine1: op> show isis route
Instance: default, AFI: ipv4, SAFI: unicast
  Prefix                    Level   Metric  Type        Next Hop                Interface
  198.51.100.100/24             1   1000000   Internal  n/a                     local
  198.51.100.30/24              1   2000000   Internal  198.51.100.21           ifl-0/1/2/12
  198.51.100.20/24              1   2000000   Internal  198.51.100.101          ifl-0/1/6/16
  198.51.100.21/24              1   2000000   Internal  198.51.100.101          ifl-0/1/6/16
  198.51.100.22/24              1   1000000   Internal  n/a                     local
  198.51.100.26/24              1   2000000   Internal  198.51.100.101          ifl-0/1/6/16
  198.51.100.41/24              1   1000000   Internal  n/a                     local
  198.51.100.42/24              1   2000000   Internal  198.51.100.21           ifl-0/1/2/12
Instance: default, AFI: ipv4, SAFI: labeled-unicast
  Prefix                    Level   Metric  SID Index   Next Hop                Interface           Label
  198.51.100.26/24              1   2000000         6   198.51.100.101          ifl-0/1/6/16    10006
  198.51.100.42/24              1   2000000         2   198.51.100.21           ifl-0/1/2/12    10002
Instance: default, AFI: ipv6, SAFI: unicast
  Prefix                    Level   Metric  Type        Next Hop                Interface
  2001:db8:0:41::/32                1   1000000   Internal  n/a                     local
  2001:db8:0:42::/32                1   2000000   Internal  2001:db8:0:3433::  ifl-0/1/2/12
```

Example 2: Summary view of the IS-IS routes for the specified instance

```
supervisor@rtbrick>spine1: op> show isis route instance default
Instance: default, AFI: ipv4, SAFI: unicast
  Prefix                    Level   Metric  Type        Next Hop                Interface
  198.51.100.100/24             1   1000000   Internal  n/a                     local
  198.51.100.30/24              1   2000000   Internal  198.51.100.21           ifl-0/1/2/12
  198.51.100.20/24              1   2000000   Internal  198.51.100.101          ifl-0/1/6/16
  198.51.100.21/24              1   2000000   Internal  198.51.100.101          ifl-0/1/6/16
  198.51.100.22/24              1   1000000   Internal  n/a                     local
  198.51.100.26/24              1   2000000   Internal  198.51.100.101          ifl-0/1/6/16
  198.51.100.41/24              1   1000000   Internal  n/a                     local
  198.51.100.42/24              1   2000000   Internal  198.51.100.21           ifl-0/1/2/12
Instance: default, AFI: ipv4, SAFI: labeled-unicast
  Prefix                    Level   Metric  SID Index   Next Hop                Interface           Label
  198.51.100.26/24              1   2000000         6   198.51.100.101          ifl-0/1/6/16    10006
  198.51.100.42/24              1   2000000         2   198.51.100.21           ifl-0/1/2/12    10002
Instance: default, AFI: ipv6, SAFI: unicast
  Prefix                    Level   Metric  Type        Next Hop                Interface
  2001:db8:0:41::/32              1   1000000   Internal  n/a                local
  2001:db8:0:42::/32              1   2000000   Internal  2001:db8:0:3433::  ifl-0/1/2/12
```

Example 3: Summary view of the IS-IS routes for the specified instance and address family (IPv4 unicast).

```
supervisor@rtbrick>spine1: op> show isis route instance default ipv4 unicast
Instance: default, AFI: ipv4, SAFI: unicast
  Prefix                    Level   Metric  Type        Next Hop                Interface
  198.51.100.100/24             1   1000000   Internal  n/a                     local
  198.51.100.30/24              1   2000000   Internal  198.51.100.21           ifl-0/1/2/12
  198.51.100.20/24              1   2000000   Internal  198.51.100.101          ifl-0/1/6/16
  198.51.100.21/24              1   2000000   Internal  198.51.100.101          ifl-0/1/6/16
  198.51.100.22/24              1   1000000   Internal  n/a                     local
  198.51.100.26/24              1   2000000   Internal  198.51.100.101          ifl-0/1/6/16
  198.51.100.41/24              1   1000000   Internal  n/a                     local
  198.51.100.42/24              1   2000000   Internal  198.51.100.21           ifl-0/1/2/12
```

Example 4: Summary view of the IS-IS routes for the specified instance and address family (IPv4 labeled-unicast).

```
supervisor@rtbrick>spine1: op> show isis route instance default ipv4 labeled-unicast
Instance: default, AFI: ipv4, SAFI: labeled-unicast
  Prefix              Level   Metric      SID Index   Next Hop        Interface       Label
  198.51.100.26/24    1   2000000         6   198.51.100.101          ifl-0/1/6/16    10006
  198.51.100.42/24    1   2000000         2   198.51.100.21           ifl-0/1/2/12    10002
```

Example 5: Summary view of the IS-IS routes for the specified instance and address family (IPv6 unicast).

```
supervisor@rtbrick>spine1: op> show isis route instance default ipv6 unicast
Instance: default, AFI: ipv6, SAFI: unicast
  Prefix                  Level   Metric     Type      Next Hop              Interface
  2001:db8:0:41::/32      1       1000000    Internal  n/a                    local
  2001:db8:0:42::/32      1       2000000    Internal  2001:db8:0:3433::   ifl-
0/1/2/12
```

IS-IS Segment Routing

Syntax:

show isis segment-routing <option>

Option	Description
global-block	Displays Segment routing global block (SRGB) information

Option	Description
global-block instance <instance>	Displays Segment routing global block (SRGB) information for the specified instance
label-binding	Displays the IS-IS segment routing label bindings information
label-binding instance <instance>	Displays the IS-IS segment routing label bindings for the specified instance
prefix-segment	Displays the IS-IS prefix segments information
prefix-segment <instance>	Displays the IS-IS prefix segments for the specified instance
adjacency-segment	Displays the IS-IS segment routing adjacency SIDs

Example 1: Summary view of the IS-IS segment routing global block

```
supervisor@rtbrick>spine1: op> show isis segment-routing global-block
Instance: default, Level: 1
  System                    SRGB Base   SRGB Range
  isr1                           1000         1000
  isr2                           2000         1000
  isr5                           5000         1000
  isr6                           6000         1000
Instance: default, Level: 2
  System                    SRGB Base   SRGB Range
  isr2                           2000         1000
  isr3                           3000         1000
  isr4                           4000         1000
  isr5                           5000         1000
```

Example 2: Summary view of the IS-IS segment routing label binding

```
supervisor@rtbrick>spine1: op> show isis segment-routing label-binding
Instance: default, Level: 1
  System                    Prefix                        Range   SID
Flags
  isr1                      198.51.100.81/24                 3      10
None
  isr2                      198.51.100.71/24                 3      20
None
Instance: default, Level: 2
  System                    Prefix                         Range   SID
Flags
  isr2                      198.51.100.81/24                  3      10
Re-advertisement
  isr5                      198.51.100.81/24                  3      10
Re-advertisement
  isr5                      198.51.100.71/24                  3      20
```

```
Re-advertisement
```

Example 3: Summary view of the IS-IS segment routing prefix segment

```
supervisor@rtbrick>spine1: op> show isis segment-routing prefix-segment
Instance: default, Level: 1
  System                    Prefix                                      SID
Flags
  isr1                      198.51.100.90/24                             100
Node
  isr1                      2001:db8:0:10::/32                           102
Node
  isr2                      198.51.100.91/24                             200
Node
  isr2                      2001:db8:0:11::/32                           202
Node
Instance: default, Level: 2
  System                    Prefix                                       SID
Flags
  isr2                      198.51.100.90/24                             100   Re-
advertisement, Node
  isr2                      198.51.100.91/24                             200
Node
  isr2                      198.51.100.92/24                             500   Re-
advertisement, Node
  isr2                      198.51.100.93/24                             600   Re-
advertisement, Node
  isr2                      2001:db8:0:10::/32                           102   Re-
advertisement, Node
```

Example 4: Summary view of the IS-IS segment routing adjacency-segment

```
supervisor@S1-STD-17-1703: cfg> show isis segment-routing adjacency-segment
Instance: ip2vrf, Level: 1
  System                       Label   Flags
  isr1                         11116   Value, Local, Persistent
  isr1                         11117   Ipv6 Encapsulation, Value, Local,
Persistent
  isr2                         11112   Value, Local, Persistent
Instance: ip2vrf, Level: 2
  System                       Label   Flags
```

IS-IS SPF

Syntax:

show isis spf <option>

Option	Description
result	Displays a summary of the IS-IS SPF results

Option	Description
result <instance>	Displays a summary of the IS-IS SPF results for the specified instance
result [level-1/level2]	Displays a summary of the IS-IS SPF results for the specified level.

Example 1: Summary view of the IS-IS SPF result

```
supervisor@rtbrick>spine1: op> show isis spf result
Instance: default, Level: 1
  Destination Node     Metric       Neighbor Node         Interface      Nexthop   Address
  1920.0000.0006.00    1000000      1920.0000.0006.00     ifl-0/1/6/16   IPv4      198.51.100.101
  1920.0000.0006.00                                       ifl-0/1/6/16   IPv6      2001:db8:0:6843::
  1920.0100.4001.00     0                                 local
  1920.0100.4002.00    1000000      1920.0100.4002.00     ifl-0/1/2/12   IPv6      2001:db8:0:3433::
                                    1920.0100.4002.00     ifl-0/1/2/12   IPv4      198.51.100.21
Instance: default, Level: 2
  Destination Node          Metric       Neighbor Node         Interface         Nexthop   Address
  1920.0100.4001.00           0                                local
```

Example 2: Summary view of the IS-IS SPF result for level-1

```
supervisor@rtbrick>spine1: op> show isis spf result level-1
Instance: default, Level: 1
  Destination Node          Metric      Neighbor Node       Interface       Nexthop   Address
  1920.0000.0006.00         1000000     1920.0000.0006.00   ifl-0/1/6/16    IPv4    198.51.100.101
                                        1920.0000.0006.00   ifl-0/1/6/16    IPv6    2001:db8:0:6843::
  1920.0100.4001.00             0                           local
  1920.0100.4002.00         1000000     1920.0100.4002.00   ifl-0/1/2/12    IPv6    2001:db8:0:3433::
                                        1920.0100.4002.00   ifl-0/1/2/12    IPv4    198.51.100.21
```

Example 3: Summary view of the IS-IS SPF result for level-2

```
supervisor@rtbrick>spine1: op> show isis spf result level-2
Instance: default, Level: 2
  Destination Node          Metric      Neighbor Node       Interface       Nexthop   Address
  1920.0100.4001.00             0                           local
```

Example 4: Summary view of the IS-IS SPF result of a specific instance for level-1

```
supervisor@rtbrick>spine1: op> show isis spf result instance default level-1
Instance: default, Level: 1
  Destination Node          Metric      Neighbor Node       Interface       Nexthop   Address
  1920.0000.0006.00         1000000     1920.0000.0006.00   ifl-0/1/6/16    IPv4    198.51.100.101
                                        1920.0000.0006.00   ifl-0/1/6/16    IPv6    2001:db8:0:6843::
  1920.0100.4001.00             0                           local
  1920.0100.4002.00         1000000     1920.0100.4002.00   ifl-0/1/2/12    IPv6    2001:db8:0:3433::
                                        1920.0100.4002.00   ifl-0/1/2/12    IPv4    198.51.100.21
```

Example 5: Summary view of the IS-IS SPF result of a specific instance for level-2

```
supervisor@rtbrick>spine1: op> show isis spf result instance default level-2
Instance: default, Level: 2
  Destination Node          Metric      Neighbor Node       Interface       Nexthop   Address
  1920.0100.4001.00              0                          local
```

IS-IS Clear Commands

Clear commands allow to reset operational states.

IS-IS Interface

Syntax:

clear isis interface <option> ...

Option	Description
statistics	Clears the statistics of all IS-IS interfaces.

Example: The example below shows how to clear IS-IS interface statistics.

```
supervisor@rtbrick>spine1: op> clear isis interface statistics
```

IS-IS Neighbor

Syntax:

clear isis neighbor <option> ...

Option	Description
neighbor	Clears neighbors of the default instance
neighbor instance <instance_name>	Clears neighbors of the specified instance
neighbor instance <instance_name> interface <interface-name>	Clears the specified interface of a specified neighbor instance

Example: The example below shows how to clear neighbors of the specified instance

```
supervisor@rtbrick>spine1: op> clear isis neighbor instance default
```

IS-IS Database

Syntax:

clear isis database <option> ...

Option	Description
Instance	Clears neighbors of the default instance
level-1	Clears area level-1
level-2	Clears area level-2
lsp	Clears IS-IS database for the specific LSP ID
System	Clears the LSDB system ID

Example: The example below shows how to clear the database instance

```
supervisor@rtbrick>spine1: op> clear isis database instance default
```

The LSDB data is cleared for instance default

2.4. OSPFv2/v3

2.4.1. OSPF Overview

OSPF (Open Shortest Path First) is an Interior Gateway Protocol that distributes routing information within a single Autonomous System (AS) in an IP network. OSPF is a link-state routing protocol that uses link-state information to form a routing table and exchange the routing information with the neighbors.

RtBrick FullStack (RBFS) supports OSPF version 2 (OSPFv2) and OSPF version 3 (OSPFv3), including authentication, LDP-IGP sync, and redistribution policy. RBFS does not support OSPFv3 Virtual Link.

OSPF routers flood LSAs (link-state advertisements) to all other routers in an autonomous system. Routers generate routing tables using the information received from the LSAs and calculate the best path to other routers in the network.

OSPF uses the Dijkstra (Shortest Path First) algorithm to calculate the best path.

LSAs contain local state information such as interfaces and the reachability of neighbors. Other routers, which receive this information as LSAs, build their LSDB (link-state database) using this information. In an OSPF network, all routers build and maintain information about the topology of that network.

OSPFv3

OSPF Version 3 (OSPFv3) is a modified version of OSPF and it provides support for the OSPF routing protocol within an IPv6 network. As such, it provides support for IPv6 addresses and prefixes. It retains most of the structure and functions in OSPFv2 (for IPv4) with a few changes.

OSPFv3 differs from OSPFv2 in the following ways:

- OSPFv3 runs on IPv6, which is based on links rather than network segments
- OSPFv3 does not depend on IP addresses
- OSPFv3 packets and the LSA format have the following changes:

  OSPFv3 router LSAs and network LSAs do not contain IP addresses, which are advertised by Type 8 LSAs and Type 9 LSAs
- In OSPFv3, information about the flooding scope is added in the LSA Type field

  OSPFv3 stores or floods unidentified packets, whereas OSPFv2 discards
- OSPFv3 supports multi-process on a link with instance ID
- OSPFv3 uses IPv6 link-local addresses for forwarding

OSPFv3 Instance ID

One of the advancements of OSPFv3 over OSPFv2 is the use of the instance ID. This instance ID is an 8-bit field within the OSPFv3 header.

The original intent for the instance ID was to support multiple instances of OSPFv3 to run on the same interface. In this way, you can manipulate which routers on a particular segment are allowed to form adjacencies. You could use an instance number of 0 through 255 to distinguish between the different OSPFv3 instances.

However, within RFC 5838, the instance ID was re-purposed to be used to support address families (AFs) with OSPFv3. The default instance of 0 is used if no other instance is defined. However, specific ranges of the instance ID map to specific AFs. According to the RFC, these ranges are:

- Instance ID 0 to 31 — IPv6 unicast AF
- Instance ID 32 to 63 — IPv6 multicast AF
- Instance ID 64 to 95 — IPv4 unicast AF
- Instance ID 96 to 127 — IPv4 multicast AF
- Instance ID 128 to 255 — Unassigned

When using IPv4 unicast or IPv6 unicast, the allowed values for the command are between 0 and 31.

RBFS only supports IPv6 unicast addresses ranging from 0 to 31.

Understanding OSPF Areas

OSPF allows for a logical partition of the autonomous system by dividing it into areas. This logical partitioning helps to limit the flooding of link-state updates within an area.

An OSPF Autonomous System can be maintained as a single-area network or can be divided as a multi-area network. In a single area AS, the topology provides link-state information of routers in the entire autonomous system.

In a multi-area AS, the topology provides the link-state information of routers belonging to that particular area, not about routers in other areas in the autonomous system. Within an area, all OSPF routers maintain separate databases which are identical.

In a multi-area OSPF network, all areas are connected to the backbone area, known as Area 0.

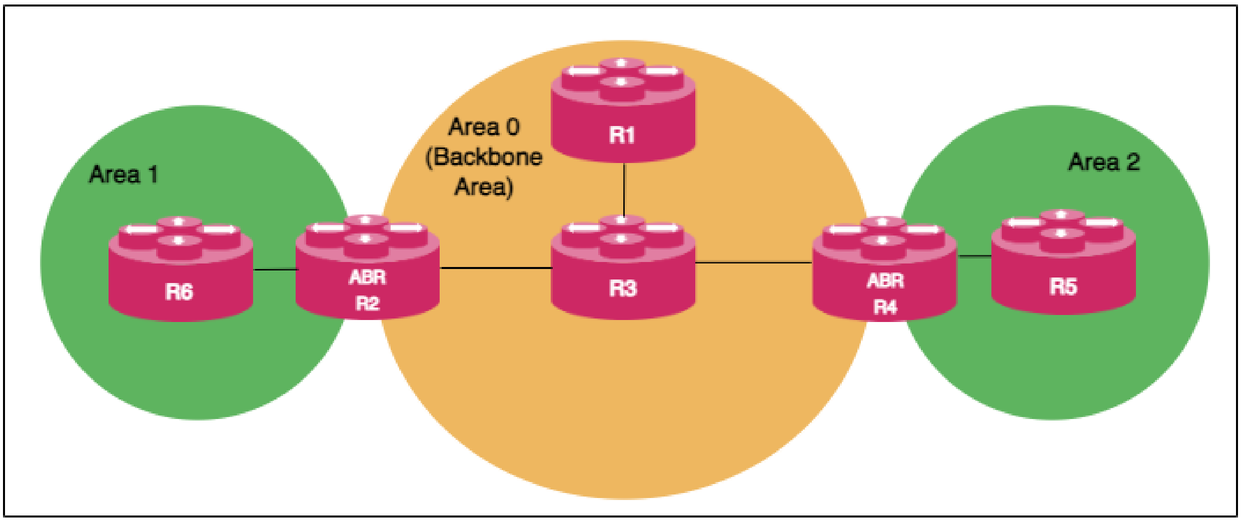
Backbone Area

The backbone area, also known as Area 0, is connected to all other areas in an OSPF network. The backbone area, which acts as a central point of communication, receives LSAs from other areas and disseminates the same to other areas.

Area Border Router

Routers that connect one or more areas with the backbone area are called Area Border Router (ABR). One interface of the ABR is connected to the backbone, while other interfaces are connected to other areas. ABRs, which belong to multiple areas in an OSPF network, maintain separate LSDBs for each area that they are connected to.

The following OSPF architectural diagram shows a simple OSPF network that is divided into areas. Area 1 and Area 2 are connected to the backbone area (Area 0) through the ABRs. Area 1 and Area 2 are not directly connected. They receive link state advertisements from each other from Area 0 which acts as the central point of communication for all other areas.



Autonomous System Boundary Router

ASBR (Autonomous System Boundary Router) serves as a gateway router to the OSPF autonomous system. ASBR can operate multiple protocols and work with other autonomous system routers that run other interior gateway protocols such as EIGRP, IS-IS, i-BGP, and so on. ASBR can import and translate different protocol routes into OSPF through the redistribution mechanism.

OSPF DR and BDR Election

An OSPF network chooses one router as a Designated Router (DR) and another as a Backup Designated Router (BDR) for a broadcast network.

DR acts as a central point of communication by receiving and distributing topology information. BDR takes over the role of DR if the DR fails. Routers in an OSPF network do not directly exchange routing information with each other. Instead, every router in the network updates routing information only with DR and BDR. DR, in turn, distributes the topology information with all other routers. This mechanism reduces network traffic significantly. OSPF chooses one router as DR and another router as BDR based on the following criteria:

- The router with the highest priority value becomes the designated router and the router with the second highest priority value becomes the BDR. You can define the priority values for routers during the interface configuration.
- If multiple routers have the same highest priority value, then the router with the highest router ID is elected as DR and the router with the second highest router ID value becomes the BDR.

You can choose a priority value from the range 0 - 255. Routers with the priority value '0' do not participate in the DR or BDR election.

Supported OSPF Standards

RBFS supports the following RFCs, which define standards for OSPFv2 and OSPFv3.

- RFC 2328, OSPF Version 2
- RFC 5340, OSPF for IPv6
- RFC 5709, OSPFv2 HMAC-SHA Cryptographic Authentication
- RFC 7166, Supporting Authentication Trailer for OSPFv3
- RFC 8665, OSPF Extensions for Segment Routing
- RFC 8666, OSPFv3 Extensions for Segment Routing

ℹ RFC and draft compliance are partial except as specified.

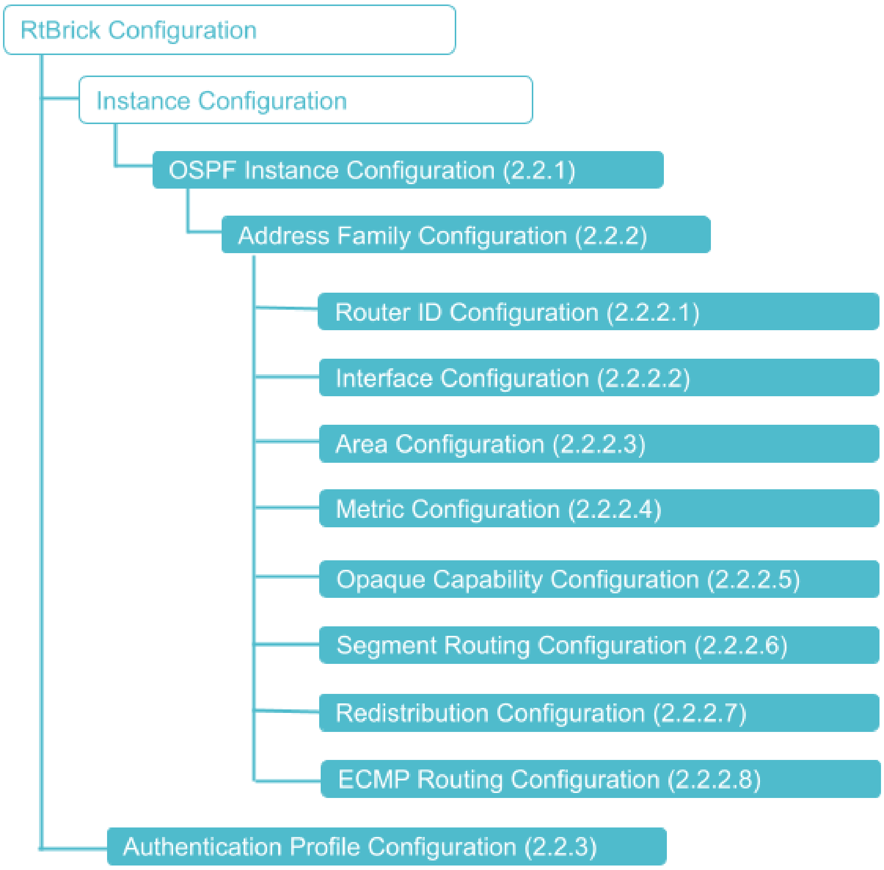
Supported Platforms

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

2.4.2. OSPF Configuration

Configuration Hierarchy

The diagram illustrates the OSPF configuration hierarchy. All OSPF configuration is performed within an instance, for example, the default instance or a VPN service instance. The OSPF instance configuration hierarchy includes parameters that are generic to the respective OSPF instance. The sub-hierarchies include parameters that are specific to redistribution or authentication.


RtBrick Configuration
Instance Configuration
OSPF Instance Configuration (2.2.1)
Address Family Configuration (2.2.2)
Router ID Configuration (2.2.2.1)
Interface Configuration (2.2.2.2)
Area Configuration (2.2.2.3)
Metric Configuration (2.2.2.4)
Opaque Capability Configuration (2.2.2.5)
Segment Routing Configuration (2.2.2.6)
Redistribution Configuration (2.2.2.7)
ECMP Routing Configuration (2.2.2.8)
Authentication Profile Configuration (2.2.3)

Configuration Syntax and Commands

The following sections describe the OSPF configuration syntax and commands.

OSPF Instance Configuration

At this configuration hierarchy, you can configure an OSPF instance.

Syntax:

set instance <instance-name> **protocol ospf**

Attribute	Description
<instance-name>	Name of the OSPF instance.

OSPF Address Family Configuration

At this configuration level, you configure the OSPF protocol address family.

> You must configure the OSPF address family on an OSPF instance before configuring other supported OSPF features.

Syntax

set instance <instance-name> **protocol ospf address-family** <afi>

Attribute	Description
<instance-name>	Name of the instance
<afi>	Specifies the Address family identifier (AFI), either IPv4 or IPv6.

The following example shows the OSPF address family (IPv4) configuration.

Example 1: OSPF Instance Address Family IPv6 Configuration

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ospf address-family ipv4
{
  "rtbrick-config:ipv4": {
    "router-id": "198.51.100.10",
    "area": [
      {
        "area-id": "0.0.0.0",
        "interface": [
          {
            "name": "ifl-0/0/0/1",
            "network-type": "p2p"
          },
          {
            "name": "lo-0/0/0/1"
          }
        ]
      }
    ]
  }
}
<...>
```

The following example shows the OSPF address family (IPv6) configuration.

Example 2: OSPF Instance Address Family IPv6 Configuration

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ospf address-family ipv6
{
  "rtbrick-config:ipv6": {
    "instance-id": [
      {
        "instance-id": 0,
        "router-id": "192.168.0.10",
        "metric": 1000,
        "segment-routing": {
          "status": "enable",
          "srgb": {
            "base": 1000,
            "range": 1000
          }
        },
        "redistribute": [
          {
            "source": "bgp",
            "metric": 2000
          },
          {
            "source": "direct",
            "policy": "ospf_policy_1"
          }
        ],
        "area": [
          {
            "area-id": "0.0.0.0",
            "metric": 1000,
            "interface": [
              {
                "name": "ifl-0/0/0/1",
                "authentication-profile": "AUTH_PROFILE_SHA_512_1"
              },
              {
                "name": "ifl-0/0/0/100",
                "metric": 20000,
                "network-type": "p2p",
                "authentication-profile": "AUTH_PROFILE_SHA_512_1"
              },
              {
                "name": "ifl-0/0/1/1",
                "metric": 40000,
                "network-type": "p2p",
                "authentication-profile": "AUTH_PROFILE_SHA_512_1"
              },
              {
                "name": "ifl-0/0/1/100",
                "metric": 30000,
                "authentication-profile": "AUTH_PROFILE_SHA_512_1"
<...>
```

OSPF Router ID Configuration

The router ID is an IP address that OSPF uses to identify a device on the network. The router ID should be configured under the address family hierarchy.

Syntax

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

Attribute	Description
<afi>	Specifies the Address family identifier (AFI), either IPv4 or IPv6.
<afi> instance-id <instance-id>	Specifies the instance ID. When using an IPv6 address family with OSPFv3, an instance ID ranging from 0 to 31 must be specified.
router-id <ipv4-address>	The router ID of the routing instance. It is recommended to specify the router ID.

Example 1: OSPF IPv4 Router Identifier Configuration

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ospf
address-family ipv4 router-id
{
  "rtbrick-config:router-id": "192.168.0.10"
}
supervisor@rtbrick>SPINE01: cfg>
```

Example 2: OSPF IPv6 Router Identifier Configuration

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ospf
address-family ipv6 instance-id 0 router-id
{
  "rtbrick-config:router-id": "192.168.0.10"
}
supervisor@rtbrick>SPINE01: cfg>
```

OSPF Hostname Configuration

Identifying routers through Open Shortest Path First (OSPF) hostnames can simplify network management as they are easier to remember than router IDs.

Syntax

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

Attribute	Description
<afi>	Specifies the Address family identifier (AFI), either IPv4 or IPv6.
<afi> instance-id <instance-id>	Specifies the instance ID. When using an IPv6 address family with OSPFv3, an instance ID ranging from 0 to 31 must be specified.
hostname <hostname>	The hostname of the routing instance.

Example 1: OSPF IPv4 Router Identifier Configuration

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ospf
address-family ipv4 hostname
{
  "rtbrick-config:hostname": "C-BNG"
}
supervisor@rtbrick>SPINE01: cfg>
```

Example 2: OSPF IPv6 Router Identifier Configuration

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ospf
address-family ipv6 instance-id 0 hostname
{
  "rtbrick-config:hostname": "C-BNG"
}
supervisor@rtbrick>SPINE01: cfg>
```

OSPF Area Configuration

A particular area is defined by its area ID.

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

Attribute	Description
<afi>	Specifies the Address family identifier (AFI), either IPv4 or IPv6.
<afi> instance-id <instance-id>	Specifies the instance ID. When using an IPv6 address family with OSPFv3, an instance ID ranging from 0 to 31 must be specified.
area <area-id>	Specifies the OSPF area ID.

Attribute	Description
area <area-id> metric	Area scope metric. Range: 1 - 65535. Default: 10000.
area <area-id> area-type stub	A stub area is an area through which or into which AS external advertisements are not flooded instead type-3 LSA advertise with a default route.
area <area-id> area-type totally-stub	Totally stub area is an area in which type-3 LSAs are not allowed instead type-3 LSAs advertise with a default route.
area <area-id> authentication-profile <authentication-profile>	Specifies the authentication profile name used to create an attachment point at the area level.
area <area-id> no-authentication-check <enable>	When enabled, OSPF packets received here will not undergo authentication validation, even if the user has enabled authentication. However, OSPF will continue to send authenticated packets from this interface.

Example 1: OSPF IPv4 Area Configuration

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ospf address-family ipv4 area 0.0.0.0
{
  "rtbrick-config:area": [
    {
      "area-id": "0.0.0.0",
      "interface": [
        {
          "name": "ifl-0/0/0/1",
          "authentication-profile": "AUTH_PROFILE_SHA_512_1"
        },
        {
          "name": "ifl-0/0/0/100",
          "metric": 20000,
          "network-type": "p2p",
          "authentication-profile": "AUTH_PROFILE_SHA_512_1"
        },
        {
          "name": "ifl-0/0/1/1",
          "metric": 40000,
          "network-type": "p2p",
          "authentication-profile": "AUTH_PROFILE_SHA_512_1"
        },
        {
          "name": "ifl-0/0/1/100",
          "metric": 30000,
          "authentication-profile": "AUTH_PROFILE_SHA_512_1"
        },
        {
          "name": "ifl-0/0/4/1",
          "metric": 60000
        },
        {
          "name": "lo-0/0/0/1"
        },
        {
          "name": "lo-0/0/0/2"
        }
      ]
    }
  ]
}
```

Example 2: OSPF IPv6 Area Configuration

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ospf address-family ipv6 instance-id 0
area 0.0.0.0
{
  "rtbrick-config:area": [
    {
      "area-id": "0.0.0.0",
      "interface": [
        {
          "name": "ifl-0/0/0/1",
          "authentication-profile": "AUTH_PROFILE_SHA_512_1"
        },
        {
          "name": "ifl-0/0/0/100",
          "metric": 20000,
          "network-type": "p2p",
          "authentication-profile": "AUTH_PROFILE_SHA_512_1"
        },
        {
          "name": "ifl-0/0/1/1",
          "metric": 40000,
          "network-type": "p2p",
          "authentication-profile": "AUTH_PROFILE_SHA_512_1"
        },
        {
          "name": "ifl-0/0/1/100",
          "metric": 30000,
          "authentication-profile": "AUTH_PROFILE_SHA_512_1"
        },
        {
          "name": "ifl-0/0/4/1",
          "metric": 60000
        },
        {
          "name": "lo-0/0/0/1"
        },
        {
          "name": "lo-0/0/0/2"
        }
      ]
    }
  ]
}
```

OSPF Interface Configuration

Enable OSPF protocol on the router interfaces.

Syntax:

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

Attribute	Description
<afi>	Specifies the Address family identifier (AFI), either IPv4 or IPv6.
<afi> instance-id <instance-id>	Specifies the instance ID. When using an IPv6 address family with OSPFv3, an instance ID ranging from 0 to 31 must be specified.
area <area-id> interface <interface-name> ldp-synchronization enable	Enables LDP OSPF Synchronization. By default, LDP OSPF synchronization is disabled.
metric <metric>	Specify the metric value of an OSPF interface.
network-type <broadcast | p2p>	broadcast - Sets the network type to broadcast; p2p - Sets the network type to point-to-point. By default, the network-type is broadcast.
router-priority <router-priority>	Sets the router priority for an interface. Allowed range: 0 - 255, Default: 1. Routers with priority value '0' do not participate in the DR or BDR election.
segment-routing index	Sets the prefix segment identifier (SID) index for the specified interface.
timer <hello | dead | wait>	Interface timer for configuring hello, dead, and wait timers. • **hello**: Sets interval time for sending hello packets to a neighbor and this time is identical on OSPF neighbor routers. Default: 10 seconds. • **dead**: Sets interval time within which if the interface does not receive any hello packet from its neighbor, the interface comes to know that the neighbor is down. Default: 40 seconds. • **wait**: Sets wait time to delay to trigger the DR/BDR election. It cannot be more than the time set for the dead interval. Default: 40 seconds.

Attribute	Description
mtu-ignore enable	If there is an MTU mismatch on both sides of the link where OSPF runs, the OSPF adjacency will not come up as the MTU value carried in the Database Description (DBD) packets. To avoid MTU validation in the Database Description (DBD) packets, configure mtu-ignore command. By default, it is disabled.
authentication <authentication-profile>	Specifies the authentication profile name used to create an attachment point at the interface level.
no-authentication-check <enable>	When enabled, OSPF packets received here will not undergo authentication validation at the interface level, even if the user has enabled authentication.

If an authentication profile is attached to an interface and an area, the authentication profile attached to the interface takes priority.

Example 1: OSPF IPv4 Interface Configuration

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ospf address-family ipv4 area 0.0.0.0
interface
{
  "rtbrick-config:interface": [
    {
      "name": "ifl-0/0/0/1",
      "authentication-profile": "AUTH_PROFILE_SHA_512_1"
    },
    {
      "name": "ifl-0/0/0/100",
      "metric": 20000,
      "network-type": "p2p",
      "authentication-profile": "AUTH_PROFILE_SHA_512_1"
    },
    {
      "name": "ifl-0/0/1/1",
      "metric": 40000,
      "network-type": "p2p",
      "authentication-profile": "AUTH_PROFILE_SHA_512_1"
    },
    {
      "name": "ifl-0/0/1/100",
      "metric": 30000,
      "authentication-profile": "AUTH_PROFILE_SHA_512_1"
    },
    {
      "name": "ifl-0/0/4/1",
      "metric": 60000
    },
    {
      "name": "lo-0/0/0/1"
    },
    {
      "name": "lo-0/0/0/2"
    }
  ]
}
```

Example 2: OSPF IPv6 Interface Configuration

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ospf address-family ipv6 instance-id 0
area 0.0.0.0 interface
{
  "rtbrick-config:interface": [
    {
      "name": "ifl-0/0/0/1",
      "authentication-profile": "AUTH_PROFILE_SHA_512_1"
    },
    {
      "name": "ifl-0/0/0/100",
      "metric": 20000,
      "network-type": "p2p",
      "authentication-profile": "AUTH_PROFILE_SHA_512_1"
    },
    {
      "name": "ifl-0/0/1/1",
      "metric": 40000,
      "network-type": "p2p",
      "authentication-profile": "AUTH_PROFILE_SHA_512_1"
    },
    {
      "name": "ifl-0/0/1/100",
      "metric": 30000,
      "authentication-profile": "AUTH_PROFILE_SHA_512_1"
    },
    {
      "name": "ifl-0/0/4/1",
      "metric": 60000
    },
    {
      "name": "lo-0/0/0/1"
    },
    {
      "name": "lo-0/0/0/2"
    }
  ]
}
```

OSPF Metric Configuration

Metric is the cost that OSPF uses to calculate and identify the best paths to other routers.

Syntax

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

Attribute	Description
<afi>	Specifies the Address family identifier (AFI), either IPv4 or IPv6.
<afi> instance-id <instance-id>	Specifies the instance ID. When using an IPv6 address family with OSPFv3, an instance ID ranging from 0 to 31 must be specified.

Attribute	Description
metric <metric>	OSPF address-family metric. Allowed range: 1 - 65535. Default: 10000. If you configure the metric at the address family, it will be applicable to the configured areas of the address-family. If you configure a metric for an area, this configured metric value will take precedence over the address-family metric configurations of this area. If you specify a metric value for an area on an interface will override any area and address-family metric configurations for this area.

Example 1: OSPF IPv4 Metric Configuration

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ospf address-family ipv4 area 0.0.0.0
metric
{
  "rtbrick-config:metric": 1000
}
```

Example 2: OSPF IPv6 Metric Configuration

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ospf address-family ipv6 instance-id 0
area 0.0.0.0 metric
{
  "rtbrick-config:metric": 1000
}
```

OSPF Opaque Capability Configuration

Enables opaque link-state advertisements. Routers in the OSPF network can receive and advertise Type-9, Type-10 and Type-11 opaque LSAs.

Syntax

set instance <instance-name> **protocol ospf address-family ipv4** <attribute> <value>

Attribute	Description
opaque-capability <enable | disable>	Enable or disable opaque LSA advertisement and reception. Set as 'enable' to enable the router to receive and advertise opaque LSAs.

Example: OSPF Opaque Capability Configuration

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ospf address-family ipv4 opaque-
capability
{
  "rtbrick-config:opaque-capability": "enable"
}
```

Segment Routing Configuration

Enable segment routing for OSPF. For configuring segment routing, you must enable the opaque capability by defining it as 'true'. For information, see the section: "Opaque Capability Configuration".

Syntax

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

Attribute	Description
<afi>	Specifies the Address family identifier (AFI), either IPv4 or IPv6.
<afi> instance-id <instance-id>	Specifies the instance ID. When using an IPv6 address family with OSPFv3, an instance ID ranging from 0 to 31 must be specified.
segment-routing srgb base <value>	Specifies the segment routing global block (SRGB) in source packet routing. SRGB is used for prefix SIDs. Supported MPLS label values are 0 - 1048575. The reserved MPLS label range is 0 - 15. In RBFS, BGP uses the label range 20000 - 100000. It is recommended to assign label values outside of these reserved ranges to avoid conflicts.
segment-routing srgb range <value>	OSPF system range of labels from the base label.

Attribute	Description
segment-routing status <disable | enable>	Enable or disable the segment routing feature. By default, the status is disabled.

Example 1: OSPF IPv4 Segment routing Configuration

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ospf address-family ipv4 segment-
routing
{
  "rtbrick-config:segment-routing": {
    "status": "enable",
    "srgb": {
      "base": 1000,
      "range": 1000
    }
  }
}
```

Example 2: OSPF IPv6 Segment routing Configuration

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ospf address-family ipv6 instance-id 0
segment-routing
{
  "rtbrick-config:segment-routing": {
    "status": "enable",
    "srgb": {
      "base": 1000,
      "range": 1000
    }
  }
}
```

OSPF Redistribution Configuration

Enable route redistribution for the routes originating from other sources or protocols such as BGP, Direct, IPoE, IS-IS, PPP, and Static.

Syntax

set instance <instance-name> **protocol ospf** <afi> <attribute> <value>

Attribute	Description
<afi>	Specifies the Address family identifier (AFI), either IPv4 or IPv6.
<afi> instance-id <instance-id>	Specifies the instance ID. When using an IPv6 address family with OSPFv3, an instance ID ranging from 0 to 31 must be specified.

Attribute	Description
redistribute <protocol>	Specifies the source protocol from which the routes are to be redistributed. The available options include BGP, Direct, IPoE, IS-IS, PPP, and Static.
metric <metric>	Specifies the metric value for the redistributed routes
metric-type <type 1 | type 2>	Specifies the external metric type for the redistributed routes.
policy	Specifies the name of the policy map. The redistribute attach point allows routes from other sources to be advertised by OSPFv2.

Example 1: OSPF IPV4 BGP Redistribution Configuration

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ospf address-family ipv4 redistribute
bgp
{
  "rtbrick-config:redistribute": [
    {
      "source": "bgp",
      "metric": 2000
    }
  ]
}
```

Example 2: OSPF IPv6 Redistribution Policy

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ospf address-family ipv6 instance-id 0
redistribute direct
{
  "rtbrick-config:redistribute": [
    {
      "source": "direct",
      "policy": "ospf_policy_1"
    }
  ]
}
```

ECMP Routing Configuration

ECMP (equal-cost multiple paths) routing is a mechanism in which routers forward packets to a destination using the multiple available best paths. This mechanism can increase network bandwidth substantially by load-balancing traffic through multiple best paths.

Syntax

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

Attribute	Description
<afi>	Specifies the Address family identifier (AFI), either IPv4 or IPv6.
<afi> instance-id <instance-id>	Specifies the instance ID. When using an IPv6 address family with OSPFv3, an instance ID ranging from 0 to 31 must be specified.
<max-load-balance>	Maximum number of equal-cost multiple paths to be calculated for load balancing. Default: 16. Allowed range: 1 - 255.

Example: OSPF IPv4 ECMP Routing Configuration

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ospf address-family ipv4 max-load-
balance
{
  "rtbrick-config:max-load-balance": 100
}
```

Example: OSPF IPv6 ECMP Routing Configuration

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ospf address-family ipv6 instance-id 0
max-load-balance
{
  "rtbrick-config:max-load-balance": 100
}
```

OSPF Authentication Configuration

OSPF supports the secure exchange of routing updates through authentication. You can enable authentication by attaching an authentication profile at the area or interface level. OSPF allows multiple keys to be attached to prevent session interruption.

The table below shows the authentication types supported by OSPFv2 and v3.

Authentication Type	OSPFv2	OSPFv3
Clear Text	Yes	No

Authentication Type	OSPFv2	OSPFv3
HMAC-SHA-1	Yes	Yes
HMAC-SHA-256	Yes	Yes
HMAC-SHA-384	Yes	Yes
HMAC-SHA-512	Yes	Yes
MD5	Yes	No

> To authenticate OSPF, there must be a global authentication profile present.

Configuring an Authentication Profile

set authentication-profile <attribute> <value>

Attribute	Description
<name>	Specifies the authentication profile name.
<name> key <key-id>	Specifies the message digest key identifier to be used by the neighboring routers for the OSPF password authentication. Allowed range: 1 - 255.
<name> key <key-id> type <auth-type>	Specifies the type of authentication that is being used, such as MD5, HMAC-SHA-1, and others.
<name> key <key-id> plain-text <text>	Specifies the password in plain text format.
<name> key <key-id> encrypted-text <text>	Specifies the password in an encrypted text format.
<name> key prefer-key-id <key-id>	Preferred key-id configuration will be used while sending out the packet with the specified key.

> - When an authentication profile is available, you can configure an authentication attachment point at the area or interface level.
> - When an authentication profile contains multiple key-IDs, and the preferred key-ID is not configured, the packet is sent using the highest key-ID.

In the example below, the authentication profile "auth-profile1" has md5, hmac-sha-1, and clear-text enabled. The preferred key-id being 20, the hmac-sha-1 method will be used for authentication.

```
    "rtbrick-config:instance": [
      {
        "name": "default",
        "address-family": [
          {
            "afi": "ipv4",
            "safi": "unicast"
          },
          {
            "afi": "ipv6",
            "safi": "unicast"
          }
        ],
        "protocol": {
          "ospf": {
            "address-family": {
              "ipv6": {
                "instance-id": [
                  {
                    "instance-id": 0,
                    "router-id": "192.168.0.10",
                    "max-load-balance": 100,
                    "metric": 1000,
                    "segment-routing": {
                      "status": "enable",
                      "srgb": {
                        "base": 1000,
                        "range": 1000
                      }
                    },
                    "redistribute": [
                      {
                        "source": "bgp",
                        "metric": 2000
                      },
                      {
                        "source": "direct",
                        "policy": "ospf_policy_1"
                      }
                    ],
                    "area": [
                      {
                        "area-id": "0.0.0.0",
                        "metric": 1000,
                        "interface": [
                          {
                            "name": "ifl-0/0/0/1",
                            "authentication-profile": "AUTH_PROFILE_SHA_512_1"
                          },
                          {
                            "name": "ifl-0/0/0/100",
                            "metric": 20000,
                            "network-type": "p2p",
                            "authentication-profile": "AUTH_PROFILE_SHA_512_1"
```

2.4.3. OSPF Operational Commands

OSPF Show Commands

OSPF Summary

Displays the OSPF protocol summary information.

Syntax:

show ospf summary <options>

Option	Description
-	Without any option, the command displays the information for all instances and address families.
instance <instance-name>	OSPF summary information for the given instance.
instance <instance-name> <afi>	OSPF summary information for the specified instance and address family. Supported AFI values are 'ipv4' and 'ipv6'.
<afi> instance <instance-name>	OSPF summary information for the specified address family and instance. Supported AFI values are 'ipv4' and 'ipv6'.

Example: OSPF summary for the default instance

```
supervisor@rtbrick>SPINE01: op> show ospf summary
Global Information:
  Neighbor State Information:
    Full       : 12
    Loading    : 0
    Exchange   : 0
    ExStart    : 0
    TwoWay     : 2
    Init       : 0
    Attempt    : 0
    Down       : 0
Instance: default, Address family: ipv4
  General information:
    Router ID: 192.168.0.10, Area count: 1, Flood interval: 1000ms
    Opaque capability: True, Segment routing capability: True
    Flags: -|-|-|-|-, Cost: 10000
    SPF initial delay: 50ms, SPF short delay: 200ms, SPF long delay: 5000ms
  Area: 0.0.0.0
    Interface count: 7
      Interface: ifl-0/0/0/1
        Address: 12.0.0.1, State: BDR, Type: broadcast, Priority: 1
        Designated router: 12.0.0.2, Backup designated router: 12.0.0.1
        Hello interval: 10s, Dead interval: 40s
        Cost: 10000, MTU: 1500
      Interface: ifl-0/0/0/100
        Address: 12.1.0.1, State: P2P, Type: p2p, Priority: 1
        Designated router: 0.0.0.0, Backup designated router: 0.0.0.0
        Hello interval: 10s, Dead interval: 40s
        Cost: 20000, MTU: 1500
      Interface: ifl-0/0/1/1
        Address: 12.2.0.0, State: P2P, Type: p2p, Priority: 1
        Designated router: 0.0.0.0, Backup designated router: 0.0.0.0
        Hello interval: 10s, Dead interval: 40s
        Cost: 40000, MTU: 1500
      Interface: ifl-0/0/1/100
        Address: 12.3.0.1, State: BDR, Type: broadcast, Priority: 1
        Designated router: 12.3.0.2, Backup designated router: 12.3.0.1
        Hello interval: 10s, Dead interval: 40s
        Cost: 30000, MTU: 1500
      Interface: ifl-0/0/4/1
        Address: 11.0.0.1, State: DROther, Type: broadcast, Priority: 1
        Designated router: 11.0.0.5, Backup designated router: 11.0.0.4
        Hello interval: 10s, Dead interval: 40s
        Cost: 60000, MTU: 1500
      Interface: lo-0/0/0/1
        Address: 192.168.0.10, State: P2P, Type: p2p, Priority: 1
        Designated router: 0.0.0.0, Backup designated router: 0.0.0.0
        Hello interval: 10s, Dead interval: 40s
        Cost: 10000, MTU:
      Interface: lo-0/0/0/2
        Address: 192.168.0.11, State: P2P, Type: p2p, Priority: 1
        Designated router: 0.0.0.0, Backup designated router: 0.0.0.0
        Hello interval: 10s, Dead interval: 40s
        Cost: 10000, MTU:
Instance: default, Address family: ipv6, Instance ID: 0
  General information:
    Router ID: 192.168.0.10, Area count: 1, Flood interval: 1000ms
    Opaque capability: True, Segment routing capability: True
    Flags: -|-|-|-|-, Cost: 10000
    SPF initial delay: 50ms, SPF short delay: 200ms, SPF long delay: 5000ms
  Area: 0.0.0.0
    Interface count: 7
      Interface: ifl-0/0/0/1
        Address: fe80::7810:99ff:fec0:0, State: BDR, Type: broadcast, Priority: 1
        Designated router: 192.168.0.20, Backup designated router: 192.168.0.10
        Hello interval: 10s, Dead interval: 40s
        Cost: 10000, MTU: 1500
      Interface: ifl-0/0/0/100
        Address: fe80::65:7810:99ff:fec0:0, State: P2P, Type: p2p, Priority: 1
        Designated router: 0.0.0.0, Backup designated router: 0.0.0.0
        Hello interval: 10s, Dead interval: 40s
        Cost: 20000, MTU: 1500
      Interface: ifl-0/0/1/1
        Address: fe80::7810:99ff:fec0:1, State: P2P, Type: p2p, Priority: 1
        Designated router: 0.0.0.0, Backup designated router: 0.0.0.0
        Hello interval: 10s, Dead interval: 40s
        Cost: 40000, MTU: 1500
      Interface: ifl-0/0/1/100
        Address: fe80::65:7810:99ff:fec0:1, State: BDR, Type: broadcast, Priority: 1
        Designated router: 192.168.0.20, Backup designated router: 192.168.0.10
        Hello interval: 10s, Dead interval: 40s
        Cost: 30000, MTU: 1500
      Interface: ifl-0/0/4/1
        Address: fe80::7810:99ff:fec0:4, State: DROther, Type: broadcast, Priority: 1
        Designated router: 192.168.0.50, Backup designated router: 192.168.0.40
        Hello interval: 10s, Dead interval: 40s
        Cost: 60000, MTU: 1500
      Interface: lo-0/0/0/1
        Address: 192:168::10, State: P2P, Type: p2p, Priority: 1
        Designated router: 0.0.0.0, Backup designated router: 0.0.0.0
        Hello interval: 10s, Dead interval: 40s
        Cost: 10000, MTU: 0
      Interface: lo-0/0/0/2
        Address: 192:168::11, State: P2P, Type: p2p, Priority: 1
        Designated router: 0.0.0.0, Backup designated router: 0.0.0.0
        Hello interval: 10s, Dead interval: 40s
        Cost: 10000, MTU: 0
<...>
```

OSPF Hostname

Displays the OSPF hostname information.

Syntax:

show ospf hostname

Example: OSPF hostname for the default instance

```
supervisor@rtbrick>SPINE01: op> show ospf hostname
Instance                       AFI   Instance ID     Router ID               Hostname
default                        ipv4                  192.168.0.10            Router1
default                        ipv6  0               192.168.0.10            Router1
```

OSPF Interface

Displays OSPF interface information.

Syntax:

show ospf interface <options>

Option	Description
-	Without any option, the command displays the interface information for all instances and address families.
detail	Displays the detailed interface information.

Option	Description
<interface-name> detail	Displays detailed information for the specified interface. Also, for the specified interface, you can display interface information with filter options: detail, ipv4, and ipv4 detail.
instance <instance-name>	OSPF interface information for the given instance. Also, for the specified instance, you can display interface information with filter options: interface name, detail, and ipv4 detail.
instance <instance-name> <interface-name>	Displays information for a specified interface for a given instance.
instance <instance-name> <interface-name> detail	Displays detailed information for a specified interface for a given instance.
<afi>	Displays OSPF interface information for the specified address family. Supported AFI values are 'ipv4' and 'ipv6'. You can display interface information for the specified address family using filters such as interface name, detail, and instance.

Example 1: OSPF interface information for the default instance

```
supervisor@rtbrick>SPINE01: op> show ospf interface
Instance: default, Address family: ipv4
  Interface            Area            IP Address                     State          Type       Cost
Priority DR              BDR             MTU
  ifl-0/0/0/1       0.0.0.0         12.0.0.1                       BDR            broadcast  10000    1
12.0.0.2        12.0.0.1        1500
  ifl-0/0/0/100     0.0.0.0         12.1.0.1                       P2P            p2p        20000    1
0.0.0.0         0.0.0.0         1500
  ifl-0/0/1/1       0.0.0.0         12.2.0.0                       P2P            p2p        40000    1
0.0.0.0         0.0.0.0         1500
  ifl-0/0/1/100     0.0.0.0         12.3.0.1                       BDR            broadcast  30000    1
12.3.0.2        12.3.0.1        1500
  ifl-0/0/4/1       0.0.0.0         11.0.0.1                       DROther        broadcast  60000    1
11.0.0.5        11.0.0.4        1500
  lo-0/0/0/1           0.0.0.0         192.168.0.10                   P2P            p2p        10000    1
0.0.0.0         0.0.0.0
  lo-0/0/0/2           0.0.0.0         192.168.0.11                   P2P            p2p        10000    1
0.0.0.0         0.0.0.0
Instance: default, Address family: ipv6, Instance ID: 0
  Interface            Area            IP Address                     State          Type       Cost
Priority DR              BDR             MTU
  ifl-0/0/0/1       0.0.0.0         fe80::7810:99ff:fec0:0         BDR            broadcast  10000    1
192.168.0.20    192.168.0.10    1500
  ifl-0/0/0/100     0.0.0.0         fe80::65:7810:99ff:fec0:0      P2P            p2p        20000    1
0.0.0.0         0.0.0.0         1500
  ifl-0/0/1/1       0.0.0.0         fe80::7810:99ff:fec0:1         P2P            p2p        40000    1
0.0.0.0         0.0.0.0         1500
  ifl-0/0/1/100     0.0.0.0         fe80::65:7810:99ff:fec0:1      BDR            broadcast  30000    1
192.168.0.20    192.168.0.10    1500
  ifl-0/0/4/1       0.0.0.0         fe80::7810:99ff:fec0:4         DROther        broadcast  60000    1
192.168.0.50    192.168.0.40    1500
  lo-0/0/0/1           0.0.0.0         192:168::10                    P2P            p2p        10000    1
```

```
0.0.0.0         0.0.0.0         0
  lo-0/0/0/2           0.0.0.0         192:168::11                    P2P            p2p        10000    1
0.0.0.0         0.0.0.0         0
```

Example 2: OSPF interface detailed information

```
supervisor@rtbrick>SPINE01: cfg> show ospf interface ifl-0/0/1/100 detail
Instance: default, Address family: ipv4
  Interface            Area            IP Address                     State          Type       Cost
Priority DR              BDR             MTU
  ifl-0/0/1/100     0.0.0.0         12.3.0.1                       BDR            broadcast  30000    1
12.3.0.2        12.3.0.1        1500
Instance: default, Address family: ipv6, Instance ID: 0
  Interface            Area            IP Address                     State          Type       Cost
Priority DR              BDR             MTU
  ifl-0/0/1/100     0.0.0.0         fe80::65:7810:99ff:fec0:1      BDR            broadcast  30000    1
192.168.0.20    192.168.0.10    1500
```

OSPF Neighbor

Displays OSPF neighbor information.

Syntax:

show ospf neighbor <options>

Option	Description
-	Without any option, the command displays the neighbor information for all instances.
instance <instance-name>	OSPF neighbor information for the given instance.
area <area-id>	OSPF neighbor information for the given area.
detail	Displays the detailed neighbor information.
interface <interface-name>	Displays the neighbor information for a specified interface.
instance <instance-name> detail	Displays detailed OSPF neighbor information for the given instance.
instance <instance-name> interface <interface-name>	Displays OSPF neighbor information for the specified interface for the given instance.
instance <instance-name> interface <interface-name> detail	Displays detailed OSPF neighbor information for the specified interface for the given instance.

Option	Description
interface <interface-name> detail	Displays detailed neighbor information for a specified interface.
<afi>	Displays OSPF neighbor information for the specified address family. Supported AFI values are 'ipv4' and 'ipv6'. You can display neighbor information for the specified address family using filters such as interface name, detail, and instance.
log	Logs neighbor event information.

Example: OSPF neighbor information for the default instance

```
supervisor@rtbrick>SPINE01: op> show ospf neighbor
Instance: default, Address family: ipv4
  Address                        Interface           Router          Area            State      Priority  DR
   BDR             Uptime          Expires
  12.0.0.2                       ifl-0/0/0/1      192.168.0.20    0.0.0.0         Full       1
12.0.0.2        12.0.0.1        0d:01h:38m:12s  33s
  12.2.0.1                       ifl-0/0/1/1      192.168.0.20    0.0.0.0         Full       1
0.0.0.0         0.0.0.0         0d:01h:38m:57s  33s
  11.0.0.3                       ifl-0/0/4/1      192.168.0.30    0.0.0.0         TwoWay     1
11.0.0.5        11.0.0.4        never           34s
  11.0.0.4                       ifl-0/0/4/1      192.168.0.40    0.0.0.0         Full       1
11.0.0.5        11.0.0.4        0d:01h:38m:11s  34s
  11.0.0.5                       ifl-0/0/4/1      192.168.0.50    0.0.0.0         Full       1
11.0.0.5        11.0.0.4        0d:01h:38m:16s  34s
  12.1.0.2                       ifl-0/0/0/100    192.168.0.20    0.0.0.0         Full       1
0.0.0.0         0.0.0.0         0d:01h:38m:57s  33s
  12.3.0.2                       ifl-0/0/1/100    192.168.0.20    0.0.0.0         Full       1
12.3.0.2        12.3.0.1        0d:01h:38m:12s  33s
Instance: default, Address family: ipv6, Instance ID: 0
  Address                        Interface           Router          Area            State      Priority  DR
   BDR             Uptime          Expires
  fe80::7845:9aff:fec0:0         ifl-0/0/0/1      192.168.0.20    0.0.0.0         Full       1
192.168.0.20    192.168.0.10    0d:01h:38m:12s  33s
  fe80::7845:9aff:fec0:1         ifl-0/0/1/1      192.168.0.20    0.0.0.0         Full       1
0.0.0.0         0.0.0.0         0d:01h:38m:52s  33s
  fe80::780d:96ff:fec0:4         ifl-0/0/4/1      192.168.0.30    0.0.0.0         TwoWay     1
192.168.0.50    192.168.0.40    never           34s
  fe80::7801:34ff:fec0:5         ifl-0/0/4/1      192.168.0.40    0.0.0.0         Full       1
192.168.0.50    192.168.0.40    0d:01h:38m:16s  34s
  fe80::7865:1aff:fec0:6         ifl-0/0/4/1      192.168.0.50    0.0.0.0         Full       1
192.168.0.50    192.168.0.40    0d:01h:38m:16s  34s
  fe80::65:7845:9aff:fec0:0      ifl-0/0/0/100    192.168.0.20    0.0.0.0         Full       1
0.0.0.0         0.0.0.0         0d:01h:38m:52s  33s
  fe80::65:7845:9aff:fec0:1      ifl-0/0/1/100    192.168.0.20    0.0.0.0         Full       1
192.168.0.20    192.168.0.10    0d:01h:38m:12s  33s
```

OSPF Segment Routing

Displays OSPF segment routing information.

Syntax:

show ospf segment-routing <options>

Option	Description
-	Without any option, the command displays the neighbor information for all instances.
global-block	Displays Segment routing global block (SRGB) information
global-block instance <instance-name>	Displays Segment routing global block (SRGB) information for the specified instance
global-block instance <instance-name> <afi>	Displays Segment routing global block (SRGB) information for the specified instance address family. Supported AFI values are 'ipv4' and 'ipv6'.
prefix-segment	Displays the OSPF prefix segment information
prefix-segment <instance> Displays the OSPF prefix segments for the specified instance and address family	prefix-segment <instance-name> <afi>

Example: OSPF neighbor information for the default instance

```
supervisor@rtbrick>SPINE01: op> show ospf segment-routing

<...>
```

OSPF Database

Displays information from OSPF link-state database that contains data about link-state advertisements (LSAs).

Syntax:

show ospf database <options>

Option	Description
advertising-router <router-id>	Displays LSDB information for the specified advertising router.
advertising-router <router-id> detail	Displays the detailed LSDB information for the specified advertising router.

Option	Description
advertising-router <router-id> ls-id <ls-id>	Displays the LSDB information for the specified link-state ID for the advertising router.
advertising-router <router-id> ls-type external	Displays the LSDB information for the specified LSA type for the advertising router. Link-state advertisement types include external, network, router, and summary.
detail	Displays detailed information from LSDB.
instance <instance-name>	Displays OSPF database information for the given instance.
ls-id <ls-id>	OSPF database information for a specific link-state identifier.
ls-type <type>	OSPF database information for the specified link-state type. Link-state advertisement types include external, network, router and summary.
area <area-id>	Displays database information for the specified OSPF area.
area <area-id> advertising-router <router-id>	Displays LSDB information for the specified advertising router for a specified OSPF area.
area <area-id> detail	Displays detailed LSDB information for the specified OSPF area.
area <area-id> ls-id <ls-id>	Displays LSDB information for the specified link-state identifier for the specified OSPF area.
area <area-id> ls-type <type>	Displays LSDB information for the specified link-state type for the specified OSPF area. Link-state advertisement types include external, network, router, and summary.
instance <instance-name> advertising router <router-id>	Displays LSDB information for the specified advertising router for the given instance.
instance <instance-name> area <area-id>	Displays LSDB information for the specified area for the given instance.

Option	Description
instance <instance-name> ls-id <ls-id>	Displays LSDB information for the specified link-state identifier for the given instance.
instance <instance-name> ls-type <type>	Displays LSDB information for the specified type of the given instance. Link-state advertisement types include external, network, router and summary.
<afi>	Displays LSDB neighbor information for the specified address family. Supported AFI values are 'ipv4' and 'ipv6'. You can display LSDB information for the specified address family using filters such as interface name, detail, and instance.

Example 1: OSPF database information for the default instance

```
supervisor@rtbrick>SPINE01: cfg> show ospf database
Instance: default, Address family: ipv4, Area: 0.0.0.0
  Type                      Link State ID   Advertising Router          Age        Sequence   Checksum
Cost
  Router                    192.168.0.10    192.168.0.10                578      0x80000008     0x262a
-
  Router                    192.168.0.20    192.168.0.20                579      0x80000009     0xddd7
-
  Router                    192.168.0.30    192.168.0.30                584      0x80000007     0x92be
-
  Network                   11.0.0.5        192.168.0.50                583      0x80000004     0xc9b1
-
  Network                   12.0.0.2        192.168.0.20                579      0x80000004     0x7962
-
  Network                   12.3.0.2        192.168.0.20                579      0x80000004     0xdc7a
-
  Network                   23.0.0.3        192.168.0.30                585      0x80000004     0x9917
-
  Summary-Network           24.0.1.0        192.168.0.20                623      0x80000004     0xbc36
15000
  Summary-Network           24.0.1.0        192.168.0.40                623      0x80000004     0x449a
15000
  Summary-Network           24.0.1.0        192.168.0.50                578      0x80000005      0x795
25000
Instance: default, Address family: ipv4
  Type                      Link State ID   Advertising Router          Age        Sequence   Checksum
Cost
  External                  200.0.1.0       192.168.0.60                619      0x80000004     0xba29
16777214
  External                  200.0.1.60      192.168.0.60                619      0x80000004     0x6047
16777214
  External                  200.0.1.61      192.168.0.60                619      0x80000004     0x5650
16777214
Instance: default, Address family: ipv6, Instance ID: 0, Area: 0.0.0.0
  Type                      Link State ID   Advertising Router          Age        Sequence   Checksum
Cost
  OSPFv3-Router             0.0.0.0         192.168.0.10                578      0x80000007     0xb041
-
  OSPFv3-Router             0.0.0.0         192.168.0.20                579      0x80000007     0x1085
-
  OSPFv3-Router             0.0.0.0         192.168.0.30                580      0x80000007      0xe89
-
  Inter-Area-Prefix         1.0.0.0         192.168.0.20                624      0x80000004     0xd4b7
15000
  Inter-Area-Prefix         1.0.0.0         192.168.0.30                624      0x80000004     0x2208
25000
  Inter-Area-Prefix         1.0.0.0         192.168.0.40                623      0x80000004     0x5c1c
```

```
15000
  Inter-Area-Prefix         1.0.0.0         192.168.0.50                623      0x80000004     0xf9ee
10000
  Intra-Area-Prefix         1.0.0.0         192.168.0.10                578      0x80000006     0x7af2
-
  Intra-Area-Prefix         1.0.0.0         192.168.0.20                579      0x80000005     0xf646
-
  Intra-Area-Prefix         1.0.0.0         192.168.0.30                580      0x80000006     0x56b2
-
  Link                      6.0.8.0         192.168.0.10                583      0x80000005     0x7d82
-
  Link                      6.0.8.0         192.168.0.20                584      0x80000005     0xa51a
-
  Link                      6.0.32.3        192.168.0.10                623      0x80000004     0x17d2
-
  Link                      6.0.32.3        192.168.0.20                624      0x80000004     0x3f6a
-
Instance: default, Address family: ipv6, Instance ID: 0
  Type                      Link State ID   Advertising Router          Age        Sequence   Checksum
Cost
  OSPFv3-External           1.0.0.0         192.168.0.60                619      0x80000004     0xa367
16777214
  OSPFv3-External           2.0.0.0         192.168.0.60                619      0x80000004     0x738a
16777214
<...>
```

Example 2: OSPF database detailed information

```
supervisor@rtbrick>SPINE01: cfg> show ospf database detail
Instance: default, Address family: ipv4, Area: 0.0.0.0 LSAs
  LSA ID: 192.168.0.10
    Advertising router: 192.168.0.10, LSA type: Router
    Sequence number: 0x80000008, Checksum: 0x262a, LSA age: 719s
    Length: 132, Options: *|-|-|-|-|-|E|*, Flags: -|-|-|-
    Number of links: 9
      Link ID: 12.0.0.2
        Link data: 12.0.0.1, Type: Transit
        Type of service: 0, Metric: 10000
      Link ID: 192.168.0.20
        Link data: 12.1.0.1, Type: P2P
        Type of service: 0, Metric: 20000
      Link ID: 12.1.0.0
        Link data: 255.255.255.252, Type: Stub
        Type of service: 0, Metric: 20000
      Link ID: 192.168.0.20
        Link data: 12.2.0.0, Type: P2P
        Type of service: 0, Metric: 40000
      Link ID: 12.2.0.0
        Link data: 255.255.255.254, Type: Stub
        Type of service: 0, Metric: 40000
      Link ID: 12.3.0.2
        Link data: 12.3.0.1, Type: Transit
        Type of service: 0, Metric: 30000
      Link ID: 11.0.0.5
        Link data: 11.0.0.1, Type: Transit
        Type of service: 0, Metric: 60000
      Link ID: 192.168.0.10
        Link data: 255.255.255.255, Type: Stub
        Type of service: 0, Metric: 10000
      Link ID: 192.168.0.11
        Link data: 255.255.255.255, Type: Stub
        Type of service: 0, Metric: 10000
  LSA ID: 192.168.0.20
    Advertising router: 192.168.0.20, LSA type: Router, Router ID: 192.168.0.20
    Sequence number: 0x80000009, Checksum: 0xddd7, LSA age: 720s
    Interface: ifl-0/0/0/100, Neighbor address: 12.1.0.2
    Length: 156, Options: *|-|-|-|-|-|E|*, Flags: -|-|-|B
    Number of links: 11
      Link ID: 12.0.0.2
        Link data: 12.0.0.2, Type: Transit
        Type of service: 0, Metric: 10000
      Link ID: 192.168.0.10
        Link data: 12.1.0.2, Type: P2P
        Type of service: 0, Metric: 20000
      Link ID: 12.1.0.0
        Link data: 255.255.255.252, Type: Stub
        Type of service: 0, Metric: 20000
<...>
```

Example 3: OSPF database for an advertising router

```
supervisor@rtbrick>SPINE01: cfg> show ospf database advertising-router 192.168.0.10
Instance: default, Address family: ipv4, Area: 0.0.0.0
  Type                      Link State ID   Advertising Router          Age        Sequence   Checksum
Cost
  Router                    192.168.0.10    192.168.0.10                791      0x80000008     0x262a
-
Instance: default, Address family: ipv6, Instance ID: 0, Area: 0.0.0.0
  Type                      Link State ID   Advertising Router          Age        Sequence   Checksum
Cost
  OSPFv3-Router             0.0.0.0         192.168.0.10                791      0x80000007     0xb041
-
  Intra-Area-Prefix         1.0.0.0         192.168.0.10                791      0x80000006     0x7af2
-
  Link                      6.0.8.0         192.168.0.10                796      0x80000005     0x7d82
-
  Link                      6.0.32.3        192.168.0.10                836      0x80000004     0x17d2
-
  Link                      6.8.8.0         192.168.0.10                836      0x80000004     0x677e
-
  Link                      6.8.32.3        192.168.0.10                836      0x80000004     0x2b9b
-
  Link                      6.32.8.0        192.168.0.10                795      0x80000006     0x3893
-
```

OSPF SPF Result

Displays SPF results.

Syntax:

show ospf spf result <options>

Option	Description
-	Without any option, the command displays the SPF result of all instances.
area <area-id>	Displays SPF result for the specified area.
instance <instance-name>	Name of the instance

Option	Description
node-id <node-id>	Displays SPF result for the specified node identifier.
area <area-id> <node-id>	Displays SPF result for the specified node identifier for a specified area.
instance <instance-name> area <area-id>	Displays SPF result for the specified area for a given instance.
instance <instance-name> node-id <node-id>	Displays SPF result for the specified node identifier for a given instance.
<afi>	Displays SPF result information for the specified address family. Supported AFI values are 'ipv4' and 'ipv6'. You can display SPF result information for the specified address family using filters such as interface name, detail, and instance.

Example 1: OSPF SPF Result for the default instance

```
supervisor@rtbrick>SPINE01: op> show ospf spf result
Instance: default, Address family: ipv4, Area: 0.0.0.0
  Node ID         Type            Cost     Advertising Router  Flags        Neighbor Node         Interface
Nexthop
  12.0.0.2        network         10000    192.168.0.20        -|-|-|-      -                     local
-
  12.3.0.2        network         30000    192.168.0.20        -|-|-|-      -                     local
-
  23.0.0.3        network         20000    192.168.0.30        -|-|-|-      192.168.0.20          ifl-0/0/0/1
12.0.0.2
  11.0.0.5        network         55000    192.168.0.50        -|-|-|-      192.168.0.20          ifl-0/0/0/1
12.0.0.2
  192.168.0.10    router          0        192.168.0.10        -|-|-|-      -                     local
-
  192.168.0.20    router          10000    192.168.0.20        -|-|-|B      192.168.0.20          ifl-0/0/0/1
12.0.0.2
  192.168.0.30    router          20000    192.168.0.30        -|-|-|B      192.168.0.20          ifl-0/0/0/1
12.0.0.2
  192.168.0.40    router          55000    192.168.0.40        -|-|-|B      192.168.0.20          ifl-0/0/0/1
12.0.0.2
  192.168.0.50    router          55000    192.168.0.50        -|-|-|B      192.168.0.20          ifl-0/0/0/1
12.0.0.2
Instance: default, Address family: ipv6, Area: 0.0.0.0, Instance ID: 0
  Node ID         Type            Cost     Advertising Router  Flags        Neighbor Node         Interface
Nexthop
  192.168.0.10    router          0        192.168.0.10        -|-|-|-      -                     local
-
  6.0.8.0         network         10000    192.168.0.20        -|-|-|-      -                     local
-
  6.8.32.3        network         30000    192.168.0.20        -|-|-|-      -                     local
-
  192.168.0.20    router          10000    192.168.0.20        -|-|-|B      192.168.0.20          ifl-0/0/0/1
fe80::7845:9aff:fec0:0
  6.0.8.0         network         20000    192.168.0.30        -|-|-|-      192.168.0.20          ifl-0/0/0/1
fe80::7845:9aff:fec0:0
  192.168.0.30    router          20000    192.168.0.30        -|-|-|B      192.168.0.20          ifl-0/0/0/1
fe80::7845:9aff:fec0:0
  192.168.0.40    router          55000    192.168.0.40        -|-|-|B      192.168.0.20          ifl-0/0/0/1
fe80::7845:9aff:fec0:0
  6.48.8.0        network         55000    192.168.0.50        -|-|-|-      192.168.0.20          ifl-0/0/0/1
fe80::7845:9aff:fec0:0
```

```
  192.168.0.50    router          55000    192.168.0.50        -|-|-|B      192.168.0.20          ifl-0/0/0/1
fe80::7845:9aff:fec0:0
```

Example 2: OSPF SPF Result for the specified node identifier for the given area

```
supervisor@rtbrick>SPINE01: op> show ospf spf result area 0.0.0.0 node-id 192.168.0.10
Instance: default, Address family: ipv4, Area: 0.0.0.0
  Node ID         Type            Cost     Advertising Router  Flags        Neighbor Node         Interface
Nexthop
  192.168.0.10    router          0        192.168.0.10        -|-|-|-      -                     local
-
Instance: default, Address family: ipv6, Area: 0.0.0.0, Instance ID: 0
  Node ID         Type            Cost     Advertising Router  Flags        Neighbor Node         Interface
Nexthop
  192.168.0.10    router          0        192.168.0.10        -|-|-|-      -                     local
-
```

OSPF SPF Log

Displays SPF Log information.

Syntax:

show ospf spf log <options>

Option	Description
-	Without any option, the command displays the SPF log of all instances.
instance <instance-name>	Displays SPF log for the specified instance.
<afi>	Displays SPF log information for the specified address family. Supported AFI values are 'ipv4' and 'ipv6'. You can display SPF log information for the specified address family using filters such as interface name, detail, and instance.

Example 1: OSPF SPF Result for the default instance

```
supervisor@rtbrick>SPINE01: op> show ospf spf log
Instance: default
  Router ID: 192.168.0.10
    Schedule timestamp: 2024-04-30 11:19:51, Area ID: 0.0.0.0, LSA type: Router
    Reason: Router LSA change, Back off timer: 50, LSA count: 1
    LS ID: 0.0.0.0, Number of schedule request: 1
    SPF start time: 2024-04-30 11:19:51, Number of nodes: 1, Number of links: 0
    Number of stub links: 0, SPF init time: 31us, SPF run time: 394us
    Router LSA change count: 1, Network LSA change count: -
    Prefix changes: -, Sequence number: 1
  Router ID: 192.168.0.10
    Schedule timestamp: 2024-04-30 11:19:51, Area ID: 0.0.0.0, LSA type: Router
    Reason: Router LSA change, Back off timer: 50, LSA count: 1
    LS ID: 192.168.0.10, Number of schedule request: 1
    SPF start time: 2024-04-30 11:19:51, Number of nodes: 1, Number of links: 0
    Number of stub links: 7, SPF init time: 7us, SPF run time: 311us
    Router LSA change count: 1, Network LSA change count: -
    Prefix changes: -, Sequence number: 2
  Router ID: 192.168.0.10
    Schedule timestamp: 2024-04-30 11:19:51, Area ID: 0.0.0.0, LSA type: Link
    Reason: Unknown map - (value) 0xa, Back off timer: 200, LSA count: 5
    LS ID: 6.0.8.0, Number of schedule request: 5
    SPF start time: 2024-04-30 11:19:51, Number of nodes: 1, Number of links: 0
    Number of stub links: 0, SPF init time: 7us, SPF run time: 172us
    Router LSA change count: 1, Network LSA change count: -
    Prefix changes: -, Sequence number: 3
  Router ID: 192.168.0.20
    Schedule timestamp: 2024-04-30 11:19:51, Area ID: 0.0.0.0, LSA type: Router
    Reason: Router LSA change, Back off timer: 5000, LSA count: 6
    LS ID: 192.168.0.20, Number of schedule request: 6
    SPF start time: 2024-04-30 11:19:56, Number of nodes: 3, Number of links: 6
    Number of stub links: 20, SPF init time: 35us, SPF run time: 182us
    Router LSA change count: 7, Network LSA change count: -
    Prefix changes: -, Sequence number: 4
<...>
```

OSPF Route

Displays OSPF routing table information.

Syntax:

show ospf route <options>

Option	Description
-	Without any option, the command displays the OSPF route information for all instances.
area <area-id>	OSPF route information for the given area.
instance <instance-name>	OSPF route information for the given instance.
instance <instance-name> <afi>	Displays OSPF route information for the specified address family and instance. Supported AFI values are 'ipv4' and 'ipv6'.
instance <instance-name> <afi> <safi>	Displays OSPF route information for the specified address family (AFI), sub-address family (SAFI), and instance. Supported AFI values are 'ipv4' and 'ipv6'. Supported SAFI values are 'unicast', and 'labeled-unicast''.

Option	Description
instance <instance-name> label <label>	Displays OSPF route information for the specified label and instance.
instance <instance-name> mpls unicast label <label> | type <type>	Displays OSPF route information for the specified MPLS unicast label or type for the instance.
prefix <ip>	Displays OSPF route information for the specified match prefix.
type	Displays information for OSPF route type. The route types include external-type-1, external-type-2, inter-area, intra-area, and ospf-direct.
<afi> <safi>	Displays OSPF route information for the specified address family (AFI) and sub-address family (SAFI). Supported AFI values are 'ipv4' and 'ipv6'. Supported SAFI values are 'unicast', and 'labeled-unicast''
label <label>	Displays information about the OSPF-labeled routes.
mpls unicast <label | type>	Displays information about OSPF MPLS routes.
area-border	Displays the OSPF Area Border Router (ABR) information. Refer to section "3.1.7. OSPF Route ABR" for the interface configuration details.
autonomous-system-boundary	Displays Autonomous System Border Router information. Refer to section "3.1.7. OSPF Route ABR" for the interface configuration details.

Example: OSPF route information for the default instance

```
supervisor@rtbrick>SPINE01: op> show ospf route
Instance: default, AFI: ipv4, SAFI: unicast
  Prefix                     Area             Type             Cost      Next Hop        Interface
  11.0.0.0/24                0.0.0.0          intra-area       55000     12.0.0.2        ifl-0/0/0/1
  12.0.0.0/23                0.0.0.0          ospf-direct      10000     n/a             local
  12.1.0.0/30                0.0.0.0          ospf-direct      20000     n/a             local
  12.2.0.0/31                0.0.0.0          ospf-direct      40000     n/a             local
  12.3.0.0/17                0.0.0.0          ospf-direct      30000     n/a             local
  23.0.0.0/24                0.0.0.0          intra-area       20000     12.0.0.2        ifl-0/0/0/1
  23.1.0.0/24                0.0.0.0          intra-area       25000     12.0.0.2        ifl-0/0/0/1
  24.0.1.0/24                0.0.0.0          inter-area       25000     12.0.0.2        ifl-0/0/0/1
  24.1.1.0/24                0.0.0.0          inter-area       20000     12.0.0.2        ifl-0/0/0/1
  25.0.1.0/24                0.0.0.0          inter-area       35000     12.0.0.2        ifl-0/0/0/1
  192.168.0.10/32            0.0.0.0          ospf-direct      10000     n/a             local
  192.168.0.11/32            0.0.0.0          ospf-direct      10000     n/a             local
  192.168.0.20/32            0.0.0.0          intra-area       20000     12.0.0.2        ifl-0/0/0/1
  192.168.0.21/32            0.0.0.0          intra-area       20000     12.0.0.2        ifl-0/0/0/1
  200.0.1.60/32                               external-type-1  16777215  12.0.0.2        ifl-0/0/0/1
  200.0.1.61/32                               external-type-1  16777215  12.0.0.2        ifl-0/0/0/1
  200.0.1.0/24                                external-type-1  16777215  12.0.0.2        ifl-0/0/0/1
  200.0.2.60/32                               external-type-1  16777215  12.0.0.2        ifl-0/0/0/1
  200.0.2.61/32                               external-type-1  16777215  12.0.0.2        ifl-0/0/0/1
  200.0.2.0/24                                external-type-1  16777215  12.0.0.2        ifl-0/0/0/1
Instance: default, AFI: ipv6, SAFI: unicast, Instance ID: 0
  Prefix                                        Area             Type             Cost      Next Hop
                 Interface
  11::/64                                       0.0.0.0          intra-area       55000
fe80::7845:9aff:fec0:0                       ifl-0/0/0/1
  12::/64                                       0.0.0.0          ospf-direct      10000     n/a
                 local
  12:1::/64                                     0.0.0.0          ospf-direct      20000     n/a
                 local
  12:2::/64                                     0.0.0.0          ospf-direct      40000     n/a
                 local
  23::/64                                       0.0.0.0          intra-area       20000
fe80::7845:9aff:fec0:0                       ifl-0/0/0/1
  24:0:1::/64                                   0.0.0.0          inter-area       25000
fe80::7845:9aff:fec0:0                       ifl-0/0/0/1
  24:1:1::/64                                   0.0.0.0          inter-area       20000
fe80::7845:9aff:fec0:0                       ifl-0/0/0/1
  25:0:1::/64                                   0.0.0.0          inter-area       35000
fe80::7845:9aff:fec0:0                       ifl-0/0/0/1
  25:1:1::/64                                   0.0.0.0          inter-area       40000
fe80::7845:9aff:fec0:0                       ifl-0/0/0/1
<...>
```

OSPF Route Area Border

Displays the OSPF Area Border Router (ABR) information.

Syntax:

show ospf route area-border

Example: OSPF Route ABR information

```
supervisor@rtbrick>SPINE01: cfg> show ospf route area-border
  192.168.0.20    10000    192.168.0.20        -|-|-|B      ifl-0/0/0/1
12.0.0.2
  192.168.0.30    20000    192.168.0.30        -|-|-|B      ifl-0/0/0/1
12.0.0.2
  192.168.0.40    55000    192.168.0.40        -|-|-|B      ifl-0/0/0/1
12.0.0.2
  192.168.0.50    55000    192.168.0.50        -|-|-|B      ifl-0/0/0/1
12.0.0.2
  192.168.0.20    10000    192.168.0.20        -|-|-|B      ifl-0/0/0/1
fe80::7845:9aff:fec0:0
  192.168.0.30    20000    192.168.0.30        -|-|-|B      ifl-0/0/0/1
fe80::7845:9aff:fec0:0
  192.168.0.40    55000    192.168.0.40        -|-|-|B      ifl-0/0/0/1
fe80::7845:9aff:fec0:0
  192.168.0.50    55000    192.168.0.50        -|-|-|B      ifl-0/0/0/1
fe80::7845:9aff:fec0:0
```

OSPF Route Autonomous System Boundary

Displays Autonomous System Boundary Router information.

Syntax:

show ospf route autonomous-system-boundary

Example: OSPF Route ASBR information

```
supervisor@rtbrick>SPINE01: cfg> show ospf route autonomous-system-boundary
Instance: default, Address family: ipv4
  Node ID         Cost     Advertising Router  Flags        Interface
Nexthop
  192.168.0.60    40000    192.168.0.20                     ifl-0/0/0/1
12.0.0.2
Instance: default, Address family: ipv6, Instance ID: 0
  Node ID         Cost     Advertising Router  Flags        Interface
Nexthop
  192.168.0.60    40000    192.168.0.20                     ifl-0/0/0/1
fe80::7845:9aff:fec0:0
```

OSPF LSA Request List

Displays the list of all link-state advertisements (LSAs) requests that have been sent or received by a router.

Syntax:

show ospf request-list <options>

Option	Description
-	Without any option, this command displays the list of all link-state advertisement (LSA) requests that have been sent from the router.
detail	Provides detailed information on the requests that have been sent from the router.
area <area-id>	OSPF request-list information for the given area.
instance <instance-name>	OSPF request-list information for the given instance.
<afi>	Displays request-list information for the specified address family. Supported AFI values are 'ipv4' and 'ipv6'.

Example 1: OSPF LSA requests sent to a neighbor

```
supervisor@rtbrick>SPINE01: op> show ospf request-list
Instance: default
  Type            Link State ID    Advertising Router          Age        Sequence   Checksum
  Summary-Network 11.0.0.0         198.51.100.20                 42      0x80000003      0x76e5
  Summary-Network 12.0.0.0         198.51.100.20                 42      0x80000003      0x603d
  Summary-Network 12.1.0.0         198.51.100.20                 42      0x80000003      0x481f
  Summary-Network 12.2.0.0         198.51.100.20                 42      0x80000003      0x4aab
  Summary-Network 12.3.0.0         198.51.100.20                 42      0x80000003      0xc5e4
  Summary-Network 23.0.0.0         198.51.100.20                 42      0x80000003      0xd5bb
  Summary-Network 23.1.0.0         198.51.100.20                 42      0x80000003      0xca2a
```

Example 2: Detailed information for OSPF LSA requests sent to a neighbor

```
supervisor@rtbrick>SPINE01: op> show ospf request-list detail
Instance: default LSAs
  LSA ID: 11.0.0.0
    Advertising router: 198.51.100.20, LSA type: Summary-Network, Router ID: 192.168.0.20
    Sequence number: 0x80000003, Checksum: 0x76e5, LSA age: 42
    Interface: ifl-0/0/0/1, Neighbor address: 25.0.1.2
    Length: 0, Options: *|-|-|-|-|-|-|*
  LSA ID: 12.0.0.0
    Advertised router: 198.51.100.20, LSA type: Summary-Network, Router ID: 192.168.0.20
    Sequence number: 0x80000003, Checksum: 0x603d, LSA age: 42
    Interface: ifl-0/0/0/1, Neighbor address: 25.0.1.2
    Length: 0, Options: *|-|-|-|-|-|-|*
  LSA ID: 12.1.0.0
    Advertised router: 198.51.100.20, LSA type: Summary-Network, Router ID: 192.168.0.20
    Sequence number: 0x80000003, Checksum: 0x481f, LSA age: 42
    Interface: ifl-0/0/0/1, Neighbor address: 25.0.1.2
    Length: 0, Options: *|-|-|-|-|-|-|*
  LSA ID: 12.2.0.0
    Advertised router: 198.51.100.20, LSA type: Summary-Network, Router ID: 192.168.0.20
    Sequence number: 0x80000003, Checksum: 0x4aab, LSA age: 42
    Interface: ifl-0/0/0/1, Neighbor address: 25.0.1.2
    Length: 0, Options: *|-|-|-|-|-|-|*

 <...>
```

OSPF Transmission List

Displays the list of all LSAs waiting to be re-sent or transmitted from the router.

Syntax:

show ospf transmit-list <option>

Option	Description
-	Without any option, this command displays the transmit list of all link-state advertisements (LSA).

Option	Description
area <area-id>	OSPF transmit-list information for the given area.
instance <instance-name>	OSPF transmit-list information for the given instance.
<afi>	Displays transmit-list information for the specified address family. Supported AFI values are 'ipv4' and 'ipv6'.

Example: OSPF LSA requests waiting to be transmitted.

```
supervisor@rtbrick>SPINE01: op> show ospf transmit-list
Instance: default, Area: 0.0.0.1, Interface: ifl-0/0/4/1, Neighbor: 25.0.1.5
  LSA ID                    LS type                   Advertising router        Transmit interval
Retransmit count
  11.0.0.0                  Summary-Network           198.51.100.20              5000                      1
  12.0.0.0                  Summary-Network           198.51.100.20              5000                      1
  23.0.0.0                  Summary-Network           198.51.100.20              5000                      1
  12.1.0.0                  Summary-Network           198.51.100.20              5000                      1
  23.1.0.0                  Summary-Network           198.51.100.20              5000                      1
  12.2.0.0                  Summary-Network           198.51.100.20              5000                      1
  12.3.0.0                  Summary-Network           198.51.100.20              5000                      1
```

OSPF Statistics

Displays OSPF statistics information.

Syntax:

show ospf statistics <options>

Option	Description
interface <interface-name>	Displays packet statistics information for the specified interface.
interface <interface-name> detail	Displays detailed packet statistics information for the specified interface.
neighbor <Neighbor-address>	Displays packet statistics information for the specified neighbor.
neighbor <Neighbor-address> <detail>	Displays detailed packet statistics information for the specified neighbor.
<afi>	Displays packet statistics information for the specified address family. Supported AFI values are 'ipv4' and 'ipv6'.

```
supervisor@rtbrick>SPINE01: op> show ospf statistics interface ifl-0/0/0/1 detail
Instance: default, Address family: ipv4
  Interface: ifl-0/0/0/1, Peer address: 12.0.0.2
    Hello packet:
      Received packets: 658, Sent packets: 659, Total errors: 0, Unsupported option: 0
      Area mismatch: 0, Area type option mismatch: 0, Dead interval mismatch: 0
      Hello interval mismatch: 0, Mask mismatch: 0, Self router ID: 0
      Obj add fail: 0, Source address mismatch: , Misc: 0
    DD packet:
      Received packets: 3, Sent packets: 4, Total errors: 1, Unsupported option: 0
      Invalid state packet rcvd: 0, MTU mismatch: 0, DD obj add fail: 0, Misc: 0, Negotiation fail: 0
      Master bit mismatch: 0, Exchange state init pkt: 0, Capabilities mismatch: 0
      Expected seq mismatch: 0, Full state init pkt: 0, Invalid lsa: 0, Invalid external lsa: 0,
      Lsdb Absent: 0, Lsa lookup fail: 0, Ls req add fail: 0
    LS request packet:
      Received packets: 0, Sent packets: 0, Total errors: 0, Invalid LSA type: 0
      Invalid state packet rcvd: 0, LSA lookup error: 0, LSA lookup fail: 0
      LSA obj add fail: 0, Misc: 0
    LS update packet:
      Received packets: 0, Sent packets: 0, Total errors: 0, Invalid LSA type: 0
      Zero length LSA: 0, LSA length exceeded: 0, LSA checksum fail: 0
      Invalid state packet rcvd: 0, LSA obj add fail: 0, Invalid LSA payload: 0, Misc: 0
    Ls ack packet:
      Received packets: 0, Sent packets: 0, Total errors: 0, LSA obj add fail: 0
      Invalid state packet rcvd: 0, Misc: 0
    Sanity errors:
      Payload max len error: 0, Payload min len error: 0, Invalid version: 0
      Invalid auth data len: 0, Auth data missing: 0, Invalid packet min len: 0
      Invalid area ID: 0, Invalid network mask: 0, Authentication fail: 1
Instance: default, Address family: ipv6, Instance ID: 0
  Interface: ifl-0/0/0/1, Peer address: fe80::7845:9aff:fec0:0
    Hello packet:
      Received packets: 658, Sent packets: 658, Total errors: 0, Unsupported option: 0
      Area mismatch: 0, Area type option mismatch: 0, Dead interval mismatch: 0
      Hello interval mismatch: 0, Mask mismatch: 0, Self router ID: 0
      Obj add fail: 0, Source address mismatch: , Misc: 0
    DD packet:
      Received packets: 4, Sent packets: 4, Total errors: 1, Unsupported option: 0
      Invalid state packet rcvd: 0, MTU mismatch: 0, DD obj add fail: 0, Misc: 0, Negotiation fail: 0
      Master bit mismatch: 0, Exchange state init pkt: 0, Capabilities mismatch: 0
      Expected seq mismatch: 0, Full state init pkt: 0, Invalid lsa: 0, Invalid external lsa: 0,
      Lsdb Absent: 0, Lsa lookup fail: 0, Ls req add fail: 0
    LS request packet:
      Received packets: 1, Sent packets: 1, Total errors: 0, Invalid LSA type: 0
      Invalid state packet rcvd: 0, LSA lookup error: 0, LSA lookup fail: 0
      LSA obj add fail: 0, Misc: 0
    LS update packet:
      Received packets: 4, Sent packets: 5, Total errors: 0, Invalid LSA type: 0
      Zero length LSA: 0, LSA length exceeded: 0, LSA checksum fail: 0
      Invalid state packet rcvd: 0, LSA obj add fail: 0, Invalid LSA payload: 0, Misc: 0
    Ls ack packet:
      Received packets: 4, Sent packets: 4, Total errors: 0, LSA obj add fail: 0
      Invalid state packet rcvd: 0, Misc: 0
    Sanity errors:
      Payload max len error: 0, Payload min len error: 0, Invalid version: 0
      Invalid auth data len: 0, Auth data missing: 0, Invalid packet min len: 0
      Invalid area ID: 0, Invalid network mask: 0, Authentication fail: 1
```

OSPF Clear Commands

Clear OSPF Neighbor

Clear OSPF neighbor state information.

Syntax:

clear ospf neighbor <options>

Option	Description
-	Without any option, the command clears all the OSPF neighbors.
instance <instance-name>	Clears OSPF neighbor information for the specified instance.
instance <instance-name> <afi>	Clears OSPF neighbor for the specified address family. Supported AFI values are 'ipv4' and 'ipv6'. Supported AFI values are 'ipv4' and 'ipv6'. When using an IPv6 address family with OSPFv3, an instance ID ranging from 0 to 31 must be specified.
instance <instance-name> <afi> area <area-id>	Clears OSPF neighbor for the specified area of the specified instance and address family.
instance <instance-name> <afi> area <area-id> interface <interface-name>	Clears OSPF neighbor for the specified interface for the specified area and address family of the specified instance.
force	Forcefully clears all the OSPF neighbors. This may impact DR/BDR election.
force instance <instance-name>	Forcefully clears the neighbor for the specified instance.
force instance <instance-name> <afi>	Forcefully clears OSPF neighbor for the specified address family.
force instance <instance-name> <afi> area <area-id>	Forcefully clears OSPF neighbor for the specified area of the specified instance and address family.
force instance <instance-name> <afi> area <area-id> interface <interface-name>	Forcefully clears OSPF neighbor for a specific interface and area of the specified instance and address family.

Example:

```
supervisor@rtbrick>SPINE01: cfg> clear ospf neighbor instance default ipv6 instance-id 0 area 0 interface
ifl-0/0/0/1
Triggered clear neighbor successfully
```

Clear OSPF Statistics

Clear the OSPF statistics for all instances or a specified instance.

Syntax:

clear ospf statistics <options>

Option	Description
-	Without any option, the command clears all the OSPF statistics.
instance <instance-name>	Clears OSPF statistics information for the specified instance.
instance <instance-name> <afi>	Clears OSPF statistics for the specified address family of the specified instance. Supported AFI values are 'ipv4' and 'ipv6'. When using an IPv6 address family with OSPFv3, an instance ID ranging from 0 to 31 must be specified.
instance <instance-name> <afi> area <area-id>	Clears OSPF statistics for the specified area of the specified instance and address family.
instance <instance-name> <afi> area <area-id> interface <interface-name>	Clears OSPF statistics for the specified interface for the specified area of the specified instance and address family.

Example:

```
supervisor@rtbrick>SPINE01: cfg> clear ospf statistics instance default ipv6 instance-id 0 area 0.0.0.0
interface ifl-0/0/0/1
Cleared statistics for all ipv6 neighbors under Instance [default] Area [0.0.0.0] Interface [ifl-0/0/0/1]
```

Clear OSPF Database

Clear the OSPF database for all instances.

Syntax:

clear ospf database <options>

Option	Description
-	Without any option, the command clears all the OSPF database information.
<afi>	Clears OSPF database for the specified address family. Supported AFI values are 'ipv4' and 'ipv6'. Supported AFI values are 'ipv4' and 'ipv6'. When using an IPv6 address family with OSPFv3, an instance ID ranging from 0 to 31 must be specified.
<afi> instance <instance-name>	Clears OSPF database for the specified area of the specified instance and address family.

Example:

```
supervisor@rtbrick>SPINE01: cfg> clear ospf database ipv6 instance default instance-id 0
Triggered clear database successfully
```

2.5. LDP

2.5.1. LDP Overview

Label distribution protocol (LDP) is the most commonly used protocol in the MPLS network. It generates and distributes labels and thus helps in MPLS packet switching and forwarding. By using LDP, label-switching routers in an MPLS network can exchange label mapping information to create label-switched paths (LSPs) for switching data packets. RtBrick FullStack (RBFS) supports Dual-stack, which means LDP can exchange FEC-label bindings over either IPv4 or IPv6 networks.

Peer Discovery

LDP sends UDP multicast hello packets to discover its neighbors and establishes neighbor adjacency with other directly connected label switch routers (LSRs). The hello message is periodically sent on LDP-enabled interfaces.

Session Establishment

After peer discovery, “initialization messages” are sent to each other. In these messages, the session Parameters are sent to each other. The LDP sessions are maintained by periodic keep-alive message.

After the LDP neighbors are discovered, the TCP session is established and the LDP FSM is triggered, and LDP session becomes operational. LSRs start exchanging label mapping information with each other.

Dual-stack LDP

By default, RBFS is dual-stack capable, which means it can exchange IPv4/IPv6 FEC bindings over IPv4/IPv6 media (LDP over IPv4/IPv6).

To enable or disable a particular address family in RBFS, use "status <enable|disable>" CLI. For details, see LDP Address Family Configuration.

When LDP is enabled on an interface that supports both IPv4 and IPv6, LDP will start exchanging IPv4 hellos. To send an IPv6 hello, the source IPv6 address must be configured. For details about configuring the source address, see LDP Instance Configuration.

By default, both IPv4 and IPv6 hello will use the same transport preference as IPv6, but this can be changed by using the "connection-preference <ipv4|ipv6>" CLI. For details, see LDP Instance Configuration.

The following points should be noted regarding this functionality:

Source address:

- Unless modified, IPv4 and IPv6 hellos will be sent with transport preference as IPv6 when the IPv6 source address is configured.
- When the IPv6 source address is not configured, only IPv4 hello will be sent with transport-preference IPv4 and still act as a dual-stack router and exchange both IPv4 and IPv6 FEC bindings.

When IPv4 status is disabled:

- Only IPv6 hello will be sent without Dual-stack TLV.
- Only IPv6 FEC binding will be exchanged.

When IPv6 status is disabled:

- Only IPv4 hello will be sent without Dual-stack TLV.
- Only IPv4 FEC binding will be exchanged.

When both IPv4 and IPv6 statuses are enabled:

- Both IPv4 and IPv6 hellos will be exchanged (IPv6 source address configuration is mandatory for sending IPv6 hello).
- By default, the hello message uses IPv6 as the transport preference, unless otherwise specified.
- Both IPv4 and IPv6 FEC bindings will be exchanged.

Label Generation

LDP generates label bindings for the IP addresses of the LDP-enabled loopback interfaces and then advertises them to all neighbors.

Label Management Modes

Label Advertisement Mode

LDP supports the Downstream Unsolicited feature in RBFS, where label bindings are advertised to all upstream neighbors. By default, label advertisement operates in the Downstream Unsolicited mode.

Label Distribution Control Mode

LDP supports the Ordered Label Distribution Control, where an LSR will initiate the transmission of the label mapping only for the prefix for which it has a label mapping from the next hop of the prefix or for which it is an egress.

Label Retention Mode

LDP supports the Liberal Label Retention Mode where all the label mapping advertisements for all routes received from all the LDP neighbors are retained.

Supported LDP Standards

RFC Number	Description
RFC 5036	LDP Specification The following modes are supported by RBFS for the features listed in RFC 5036: • Label advertisement: Downstream Unsolicited mode is supported but not Downstream on Demand mode. • Label distribution control: Ordered mode is supported, but not Independent mode. • Label retention: Liberal mode is supported, but not Conservative mode.
RFC 5283	LDP Extension for Inter-Area Label Switched Paths (LSPs)
RFC 5443	LDP IGP Synchronization
RFC 7552	IPv6 Dual-Stack

ℹ RFC and draft compliance are partial except as specified.

Supported LDP Features

The following LDP features are supported in this release of RBFS:

- Support for the following label management modes.

  Downstream unsolicited mode in label advertisement

  Ordered mode in the label distribution control

  Liberal mode in label retention
- Loop detection
- Inter-area support
- Tracking IGP metric
- IGP LDP synchronization
- LDP Dual-stack support
- LDP TCP authentication
- LDP redistribution

- LDP policy configuration

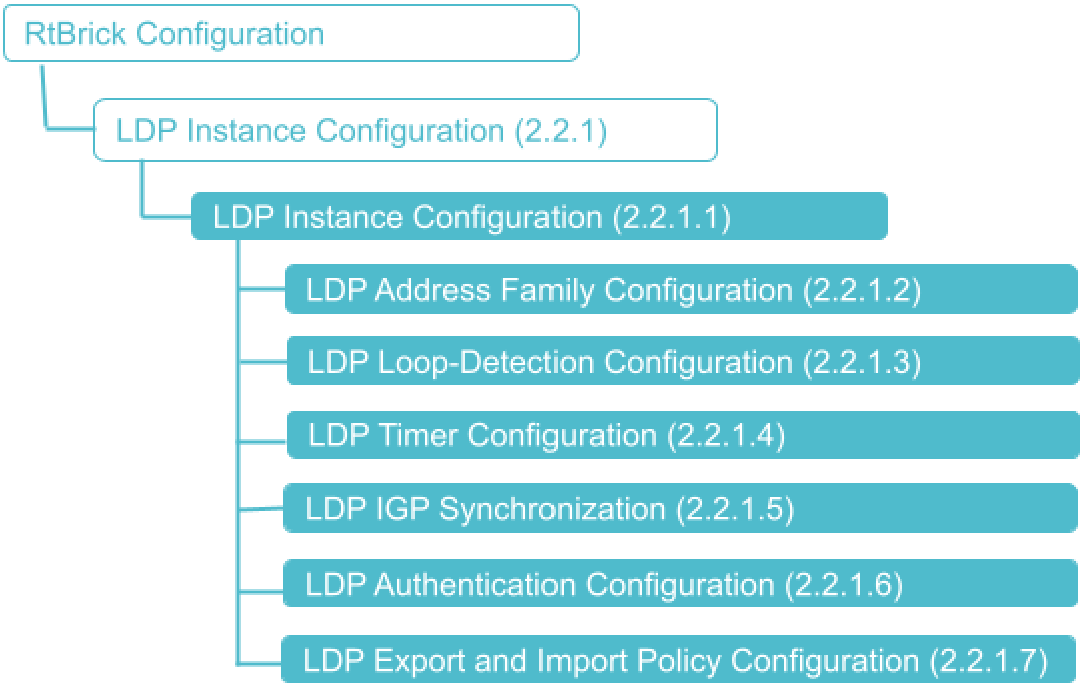
Supported Platforms

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

2.5.2. LDP Configuration

LDP Configuration Hierarchy

The diagram below illustrates the LDP configuration hierarchy.


RtBrick Configuration
LDP Instance Configuration (2.2.1)
LDP Instance Configuration (2.2.1.1)
LDP Address Family Configuration (2.2.1.2)
LDP Loop-Detection Configuration (2.2.1.3)
LDP Timer Configuration (2.2.1.4)
LDP IGP Synchronization (2.2.1.5)
LDP Authentication Configuration (2.2.1.6)
LDP Export and Import Policy Configuration (2.2.1.7)

Configuration Syntax and Commands

The following sections describe the LDP configuration syntax and commands.

LDP Instance Configuration

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

Syntax:

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

Attribute	Description
<instance-name>	Name of the LDP instance.
interface <name>	Name of the logical interface.
router-id <router-id>	Router identifier in IPv4 format.
address-family <afi>	Address family identifier (AFI). Supported values: ipv4, or ipv6. Refer to section 2.5.2.2.2, “LDP Address Family Configuration” for LDP address family configuration details.
connection-preference <afi>	Specifies the connection preference for the TCP session. Supported values: ipv4, or ipv6. By default, IPv6 is used as the preferred TCP connection if an IPv6 source address is configured. Refer to section Dual-stack LDP for more information on the LDP Dual-stack behaviour.
igp-synchronization <…>	LDP IGP synchronization configuration. This option is supported only on interfaces running Intermediate System-to-System (IS-IS) or OSPFv2 processes. Refer to section 2.5.2.2.5, “LDP IGP Synchronization” for LDP-IGP synchronization configuration details.
source-address <ipv4|ipv6> <source-address>	Use the specified IP addresses (IPv4 or IPv6) as the transport address for the LDP session. For LDP over IPv6, the IPv6 source address is mandatory. Refer to section Dual-stack LDP for more information on the LDP Dual-stack behavior.
loop-detection <…>	The LDP loop detection feature enables LDP to detect loops during an LSP establishment. Refer to section 2.5.2.2.3, “LDP Loop Detection Configuration” for the loop detection configuration details.
timer <…>	Specifies the Hello hold time, Hello interval, Keepalive hold time, and Keepalive interval. Refer to section 2.5.2.2.4, “LDP Timer Configuration” for the LDP timer configuration details.

Attribute	Description
peer <ipv4|ipv6> <address> authentication-id <...>	Specifies an IPv4 or IPv6 LDP peer attributes to apply TCP authentication. Refer to section 2.5.2.2.6, “LDP Authentication Configuration”.
peer <ipv4|ipv6> <address> export-policy|import-policy <...>	Specifies an IPv4 or IPv6 LDP peer attributes to apply import/export policy configurations. Refer to section 2.5.2.2.7, “LDP Import and Export Policy Configuration”.

Example: LDP Instance Configuration

The following example shows some LDP instance configuration attributes. The further LDP configurations like timers and loop detection are shown in the examples in the subsequent sections.

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ldp
{
    "rtbrick-config:ldp": {
      "router-id": "198.51.100.1",
      "interface": [
        {
          "name": "ifl-0/0/0/1"
        },
        {
          "name": "ifl-0/0/0/100"
        },
        {
          "name": "ifl-0/0/0/101"
        },
        {
          "name": "ifl-0/0/1/102"
        },
        {
          "name": "ifl-0/0/2/1"
        },
        {
          "name": "ifl-0/0/3/1"
        },
        {
          "name": "lo-0/0/0/1"
        },
        {
          "name": "lo-0/0/0/2"
        },
        {
          "name": "lo-0/0/0/3"
        },
        {
          "name": "lo-0/0/0/4"
        },
        {
          "name": "lo-0/0/0/5"
        }
      ]
    }
  }
supervisor@rtbrick>SPINE01: cfg>
```

LDP Address Family Configuration

The address-family command allows you to enable the address families that LDP will route and configure settings that are specific to that address family.

Syntax:

set instance <instance-name> **protocol ldp address-family** <attribute> <value>

Attribute	Description
<afi>	Address family identifier (AFI). Supported values: ipv4, ipv6
<afi> status <enable|disable>	Enable or disable address family. By default, both IPv4 and IPv6 address families are enabled, as LDP supports dual stack. Refer to section Dual-stack LDP for more information on the LDP Dual-stack behavior.
<afi> redistribute <source>	Specifies the source from which the routes are to be redistributed. The available options include direct, ipoe, isis, ospf, ppp, and static.
<afi> redistribute <source> policy <policy>	Specifies the name of the policy map. The redistribute attach point allows routes from other sources to be advertised by LDP. The policy can be applied only to the routes that are redistributed from other sources to LDP.

Example 1: LDP Address Family Configuration

```
supervisor@rtbrick>SPINE01: cfg>  show config instance default protocol ldp
address-family
{
  "rtbrick-config:address-family": [
    {
      "afi": "ipv6",
      "status": "disable"
    }
  ]
}
```

```
supervisor@rtbrick>SPINE01: cfg>
```

Example 2: LDP Redistribution Configuration

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ldp
address-family ipv4 redistribution direct
{
  "rtbrick-config:redistribution": [
    {
      "source": "direct"
    }
  ]
}
supervisor@rtbrick>SPINE01: cfg>
```

Example 3: LDP Policy Configuration

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ldp
address-family ipv4 redistribution
{
  "rtbrick-config:redistribution": [
    {
      "source": "direct",
      "policy": "filter-link-addres"
    }
  ]
}
supervisor@rtbrick>SPINE01: cfg>
```

LDP Loop Detection Configuration

The LDP loop detection feature enables LDP to detect loops during an LSP establishment.

Syntax:

set instance <instance-name> **protocol ldp loop-detection** <attribute> <value>

Attribute	Description
hop-count <hop-count>	Specifies the hop count limit for loop detection. Range: 0-255. Default: 32.
status <enable|disable>	Enables or disables loop detection. By default, this option is disabled. When this option is enabled, both hop count and path vector are enabled.

Attribute	Description
vector-length <vector-length>	Specifies the path vector length limit for loop detection. Range: 0-255. Default: 32.

Example 1: LDP Loop Detection Configuration

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ldp loop-
detection
{
    "rtbrick-config:loop-detection": {
      "enable": "true",
      "hop-count": 64,
      "vector-length": 64
    }
  }
supervisor@rtbrick>SPINE01: cfg>
```

LDP Timer Configuration

Specify the hello timer and hold-down timer for LDP adjacency. Similarly, specify the keepalive and keepalive timeout settings for the LDP session.

Syntax:

set instance <instance-name> **protocol ldp timer** <attribute> <value>

Attribute	Description
hello hold-time <hold-time>	Specifies the hello hold-time interval in seconds before declaring a neighbor to be down. Range: 0-65535. Default: 15.
hello interval <interval>	Specifies the hello messages interval in seconds. Range: 0-65535. Default: 5.
session keepalive-interval <keepalive-interval>	Specifies the session keepalive messages interval in seconds. Range: 1-65535. Default: 10.
session keepalive-timeout <keepalive-timeout>	Specifies the session keepalive timeout in seconds before declaring a session to be down. Range: 1-65535. Default: 30.

Example 1: LDP Timer Configuration

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ldp timer
```

```
{
    "rtbrick-config:timer": {
      "hello": {
        "interval": 10,
        "hold-time": 20
      },
      "session": {
        "keepalive-interval": 3000,
        "keepalive-timeout": 5000
      }
    }
  }
supervisor@rtbrick>SPINE01: cfg>
```

LDP IGP Synchronization

Synchronization between LDP and the underlying interior gateway protocol (IGP) ensures that the LDP path is fully established before the IGP path is used for forwarding traffic. LDP IGP synchronization is supported only on interfaces running Intermediate System-to-System (IS-IS) or OSPFv2 processes.

Syntax:

set instance <instance-name> **protocol ldp igp-synchronization** <attribute> <value>

Attribute	Description
hold-timer <hold-timer>	Specifies the hold-timer in seconds to limit how long the IGP session must wait before declaring the LDP synchronization. Range: 0-60. Default: 10.

Example 1: LDP IGP Synchronization Configuration

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ldp igp-
synchronization
{
    "rtbrick-config:igp-synchronization": {
     hold-timer": 60,
    }
  }
supervisor@rtbrick>SPINE01: cfg>
```

LDP Authentication Configuration

To meet the security requirements of LDP sessions, configure LDP authentication.

Syntax:

set instance <instance-name> **protocol ldp peer ipv4|ipv6** <address> **authentication-id** <authentication-id>

Attribute	Description
<address>	Specifies the transport IP address of the peer.
<authentication-id>	Authentication Tuple Identifier

Example 1: LDP Authentication Configuration

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ldp peer
ipv4
{
  "rtbrick-config:ipv4": [
    {
      "address": "192.168.1.2",
      "authentication-id": "auth_id_1"
    }
  ]
}
supervisor@rtbrick>SPINE01: cfg>
```

TCP Authentication Configuration

In the instance TCP authentication hierarchy, you can optionally enable MD5 or HMAC SHA authentication. Authentication is not configured for LDP directly, but for the TCP sessions used by LDP.

Syntax:

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

Attribute	Description
<authentication-id>	Authentication identifier
type <type>	Authentication identifier such as MD5
key1-id <key1-id>	Key ID1 of the receiver
key1-encrypted-text <key1-encrypted-text>	Encrypted text of key1

Attribute	Description
key1-plain-text <key1-plain-text>	Plain text of key1
key2-id <key2-id>	Key ID2 of the receiver
key2-encrypted-text <key2-encrypted-text>	Encrypted text of key2
key2-plain-text <key2-plain-text>	Plain text of key2

Example: LDP TCP Authentication Configuration

```
supervisor@rtbrick>SPINE01: cfg> show config instance default tcp authentication
auth_id_1
{
  "rtbrick-config:authentication": [
    {
      "authentication-id": "auth_id_1",
      "type": "MD5",
      "key1-id": 1,
      "key1-encrypted-text": "$2a6fd7db50a18a9f1f16b5c5b4214fab0"
    }
  ]
}
supervisor@rtbrick>SPINE01: cfg>
```

LDP Import and Export Policy Configuration

Syntax:

set instance <instance-name> **protocol ldp peer ipv4|ipv6** <address> <attribute> <value>

Attribute	Description
<address>	Specifies the IPv4 or IPv6 address.
export-policy <export-policy>	Export policy identifier
import-policy <import-policy>	Import policy identifier

Example 1: LDP Export Configuration

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ldp peer
ipv4
{
  "rtbrick-config:ipv4": [
    {
      "address": "192.168.1.2",
      "export-policy": "exp-policy1"
    }
  ]
}
supervisor@rtbrick>SPINE01: cfg>
```

Example 3: LDP Import Configuration

```
supervisor@rtbrick>SPINE01: cfg> show config instance default protocol ldp peer
ipv4
{
  "rtbrick-config:ipv4": [
    {
      "address": "192.168.1.2",
      "import-policy": "imp-policy1"
    }
  ]
}
supervisor@rtbrick>SPINE01: cfg>
```

2.5.3. LDP Operational Commands

LDP Show Commands

The LDP show commands provide detailed information about the LDP protocol operations.

LDP Summary

Syntax:

show ldp summary <options>

Option	**Description**
-	Without any option, the command displays the LDP summary information for all instances.
instance <instance-name>	Displays LDP summary information about the specified instance.

Example: LDP summary for the default instance

```
supervisor@rtbrick>SPINE01: op> show ldp summary
Instance: default
  General information:
    LDP identifier: 198.51.100.1:0, Version: 1
    FEC resolution: Best match
    Protocol preference: 9
    LSR ID: 198.51.100.1
    IPv4 Status: True
    IPv6 Status: True
  Modes:
    Advertisement mode: Downstream Unsolicited
    Advertisement control mode: Ordered
    Label retention mode: Liberal
  Capabilities:
    IPv6 address family: -  , Graceful restart: False
    Loop detection: False
      Hop count: -, Vector length: -
  Timers:
    Adjacency:
      Hello: 5s, Holdtime: 15s
    Targeted adjacency:
      Hello: 15s, Holdtime: 45s
    Session:
      Keepalive: 10s, Holdtime: 30s
  Statistics:
    Adjacency:
      Link adjacency: 5, Targeted adjacency: 0
    Session:
      Session in non-existent: 0, Session in initialized: 0
      Session in opensent: 0, Session in openconfirm: 0
      Session in operational: 2
supervisor@rtbrick>SPINE01: op>
```

LDP Neighbor

Syntax:

show ldp neighbor <options>

Option	Description
-	Without any option, this command displays information about LDP neighbors.
detail	Detailed information about the LDP neighbors.
instance <instance-name>	Displays LDP neighbor information about the specified instance.
instance <instance-name> detail	Displays detailed LDP neighbor information about the specified instance.

Option	Description
instance <instance-name> ldp-id <ldp-id>	Displays LDP neighbor information about the specified LDP identifier and instance.
interface <name>	Displays LDP neighbor information about the specified interface.
interface <name> detail	Displays detailed LDP neighbor information about the specified interface.
ldp-id <ldp-id>	Displays LDP neighbor information about the specified LDP identifier.

Example 1: Summary view of LDP Neighbor

```
supervisor@rtbrick>SPINE01: op> show ldp neighbor
Instance: default
  Interface           LDP ID              Transport IP  Up Since
Expires
  ifl-0/0/0/1      198.51.100.2:0      198.51.100.2   Thu Feb 09 12:17:15       in
11s
  ifl-0/0/2/1      198.51.100.3:0      198.51.100.3   Thu Feb 09 12:17:31       in
12s
  ifl-0/0/0/100    198.51.100.2:0      198.51.100.2   Thu Feb 09 12:17:15       in
11s
  ifl-0/0/0/101    198.51.100.2:0      198.51.100.2   Thu Feb 09 12:17:15       in
11s
  ifl-0/0/1/102    198.51.100.2:0      198.51.100.2   Thu Feb 09 12:17:15       in
11s
supervisor@rtbrick>SPINE01: op>
```

Example 2: Detailed View of LDP Neighbor

```
supervisor@rtbrick>SPINE01: op> show ldp neighbor detail
Instance: default
  LDP-Identifier: 198.51.100.2:0, Interface: ifl-0/0/0/1, Type: Link
  Negotiated holdtime: 15000, Expiry time: 13s 183407us
  Local link address: 192.0.2.1, Peer link address: 192.0.2.2
  Local transport address: 198.51.100.1:0, Peer transport address: 198.51.100.2
  Local holdtime: 15, Peer holdtime: 15, Up since: Tue May 02 13:28:17
  Local transport preference : ipv4, Peer transport preference : ipv4
  Last transition time: Tue May 02 13:38:52 GMT +0000 2023
<...>
```

LDP Interface

Syntax:

show ldp interface <option>

Option	Description
-	Without any option, this command displays information on all configured LDP interfaces.
<interface-name>	Specify the name of the interface.
detail	Detailed interface information.

Example 1: Specified LDP interface details

```
supervisor@rtbrick: cfg> show ldp interface ifl-0/0/1/1
Instance: green
  Interface: ifl-0/0/1/1
    Primary IPv4 Address: 12.1.0.1, Primary IPv6 Address: fe80::783f:e5ff:fec0:1
    Session Hold: 30, Session Keepalive: 10
    Discovery Hello: 5, Discovery Hold: 15
    Neighbor count: 1, Transport preference: ipv4
    IPv4 enable: True, IPv6 enable: True, Cisco interop: False
```

Example 2: View of LDP interfaces

```
supervisor@rtbrick: cfg> show ldp interface
Instance: default
  Interface             Primary IPv4 Address      Primary IPv6 Address
  ifl-0/0/0/1           12.0.0.1                  fe80::783f:e5ff:fec0:0
  ifl-0/0/0/100         12.0.1.1                  fe80::65:783f:e5ff:fec0:0
  ifl-0/0/0/101         12.0.2.1                  fe80::66:783f:e5ff:fec0:0
  ifl-0/0/1/102         12.0.3.1                  fe80::67:783f:e5ff:fec0:1
  ifl-0/0/2/1           13.0.0.1                  fe80::783f:e5ff:fec0:2
  ifl-0/0/3/1           14.0.0.1                  fe80::783f:e5ff:fec0:3
  lo-0/0/0/1            192.168.0.1
  lo-0/0/0/2            192.168.1.1
  lo-0/0/0/3            192.168.2.1
  lo-0/0/0/4            192.168.3.1
  lo-0/0/0/5            192.168.4.1
Instance: green
  Interface             Primary IPv4 Address      Primary IPv6 Address
  ifl-0/0/0/202         12.1.3.1                  fe80::cb:783f:e5ff:fec0:0
  ifl-0/0/1/1           12.1.0.1                  fe80::783f:e5ff:fec0:1
  ifl-0/0/1/200         12.1.1.1                  fe80::c9:783f:e5ff:fec0:1
  ifl-0/0/1/201         12.1.2.1                  fe80::ca:783f:e5ff:fec0:1
  ifl-0/0/2/200         13.1.0.1                  fe80::c9:783f:e5ff:fec0:2
  ifl-0/0/3/200         14.1.0.1                  fe80::c9:783f:e5ff:fec0:3
  lo-0/0/1/1            172.168.0.1
  lo-0/0/1/2            172.168.1.1
  lo-0/0/1/3            172.168.2.1
  lo-0/0/1/4            172.168.3.1
  lo-0/0/1/5            172.168.4.1
```

LDP Session

Syntax:

show ldp session <options>

Option	Description
-	Without any option, this command displays a summary of LDP session information.
detail	Displays detailed information about the LDP sessions.
instance <instance-name>	Displays LDP session information about the specified instance.
instance <instance-name> detail	Displays detailed LDP session information about the specified instance.
instance <instance-name> ldp-id <ldp-id>	Displays LDP session information about the specified LDP identifier and instance.
ldp-id <ldp-id>	Displays LDP session information about the specified LDP identifier.

Example 1: Summary view of LDP Session

```
supervisor@rtbrick>SPINE01: op> show ldp session
Instance: default
  LDP ID            Peer IP           State             Up/Down           FECRcvd   FECSent
  198.51.100.2:0    198.51.100.2       Operational       0d:00h:29m:44s        15        15
  198.51.100.3:0    198.51.100.3       Operational       0d:00h:29m:29s        15        15
supervisor@rtbrick>SPINE01: op>
```

Example 2: Detailed View of LDP Session

```
supervisor@rtbrick>SPINE01: op> show ldp session detail
Instance: default
  LDP Identifier: 198.51.100.2:0, Peer IP: 198.51.100.2, Local IP: 198.51.100.1
    Type: link, State: Operational, Uptime: 0d:00h:34m:35s
    Reason:
    Last transition: Thu Feb 09 12:17:28 GMT +0000 2023, Flap count: 0
  Advertisement Mode:
    Peer: Downstream unsolicited, Local: Downstream unsolicited
    Negotiated: Downstream unsolicited
  Timers:
    Connect retry: 10s
    Peer keepalive interval: 10s, Local keepalive interval: 10s
    Peer keepalive timeout: 30s, Local keepalive timeout: 30s
    Negotiated keepalive interval: 10s
    Negotiated keepalive timeout: 30s
  Received messages:
    Initialization: 1, KeepAlive: 208, Notification: 0
    Address: 1, Address Withdraw: 0, Label Mapping: 15
    Label Withdraw: 0, Label Release: 0
  Sent messages:
    Initialization: 1, KeepAlive: 208, Notification: 0
    Address: 1, Address Withdraw: 0, Label Mapping: 15
    Label Withdraw: 0, Label Release: 0
  Capability:
    Typed WildCard FEC:
    Local Support: True, Peer Support: True, Negotiated: True
  Total received messages:
    Initialization: 1, KeepAlive: 92, Notification: 0
    Address: 2, Address Withdraw: 0, Label Mapping: 20
  Total sent messages:
    Initialization: 1, KeepAlive: 92, Notification: 0
    Address: 2, Address Withdraw: 0, Label Mapping: 20
    Label Withdraw: 0, Label Release: 0
<...>
```

LDP Address

Syntax:

show ldp address <options>

Option	Description
-	Without any option, this command displays a summary of all the interface addresses received from the LDP sessions.
instance <instance-name>	Displays LDP address information about the specified instance.
instance <instance-name> <afi>	Displays LDP address of the specified address family (AFI). Supported values: ipv4, ipv6.
instance <instance-name> ldp-id <ldp-id>	Displays LDP address information about the specified LDP identifier and instance.
ldp-id <ldp-id>	Displays LDP address information about the specified LDP identifier.

Example: Summary View of LDP Address

```
supervisor@rtbrick>SPINE01: op> show ldp address
Instance: default, LDP Identifier: 198.51.100.2:0, AFI: ipv4
  198.51.100.61
  198.51.100.102
  198.51.100.63
```

```
    198.51.100.94
    198.51.100.2
    198.51.100.65
    198.51.100.222
    198.51.100.21
    198.51.100.2145
    198.51.100.48
  <...>
```

LDP Binding

Syntax:

show ldp binding <options>

Option	Description
-	Without any option, this command displays a summary of all the LDP label bindings.
instance <instance-name>	Displays LDP label binding information about the specified instance.
instance <instance-name> prefix <ip>	Displays LDP label binding information about the specified prefix and instance. Supported prefix values: ipv4, ipv6.
prefix <ip>	Displays the LDP label binding information for the specified prefix. Supported prefix values: ipv4, ipv6.
received	Displays the LDP received label binding information of the LDP sessions.
received instance <instance-name>	Displays LDP received label binding information of the specified instance.
received instance <instance-name> ldp-id <ldp-id>	Displays LDP received label binding information about the specified LDP identifier and instance.
received ldp-id <ldp-id>	Displays LDP received label binding information of the specified LDP identifier.
sent	Displays the LDP sent label binding information of the LDP sessions.
sent instance <instance-name>	Displays LDP sent label binding information of the specified instance.

Option	Description
sent instance <instance-name> ldp-id <ldp-id>	Displays LDP sent label binding information about the specified LDP identifier and instance.
sent ldp-id <ldp-id>	Displays LDP sent label binding information of the specified LDP identifier.

Example 1: Summary view of LDP Binding

```
supervisor@rtbrick>SPINE01: op> show ldp binding

Instance: default, AFI: ipv4
  Prefix                In Label           Out Label          LDP ID
Status
  198.51.100.1/32        -                  label:3            -
Best
                        label:20066        -                  198.51.100.3:0
Non-best
                        label:20065        -                  198.51.100.2:0
Non-best
  198.51.100.11/32        -                  label:3            -
Best
                        label:20066        -                  198.51.100.3:0
Non-best
                        label:20065        -                  198.51.100.2:0
Non-best
  198.51.100.41/32        -                  label:3            -
Best
                        label:20066        -                  198.51.100.3:0
Non-best
                        label:20065        -                  198.51.100.2:0
Non-best
  198.51.100.44/32        -                  label:3            -
Best
                        label:20066        -                  198.51.100.3:0
Non-best
                        label:20065        -                  198.51.100.2:0
Non-best
  198.51.100.47/32        -                  label:3            -
Best
                        label:20066        -                  198.51.100.3:0
Non-best
                        label:20065        -                  198.51.100.2:0
Non-best
  198.51.100.2/32        label:3            label:20065       198.51.100.2:0
Best
                        label:20065        -                  198.51.100.3:0
Non-best
  198.51.100.21/32        label:3            label:20065      198.51.100.2:0
Best
                        label:20065        -                  198.51.100.3:0
Non-best
  198.51.100.42/32        label:3            label:20065      198.51.100.2:0
Best
                        label:20065        -                  198.51.100.3:0
Non-best
```

```
<...>
```

Example 2: Summary view of LDP Binding for the specified prefix

```
supervisor@rtbrick>SPINE01: op> show ldp binding prefix 198.51.100.2/32
Instance: default, AFI: ipv4
  Prefix                In Label           Out Label          LDP ID
Status
  198.51.100.2/32        label:3            label:20065       198.51.100.2:0
Best
                        label:20065        -                  198.51.100.3:0
Non-best
supervisor@rtbrick>SPINE01: op>
```

LDP Route

Syntax:

show ldp route <options>

Option	Description
-	Without any option, this command displays a summary of LDP route information.
instance <instance-name>	Displays LDP route information for the specified instance.
instance <instance-name> <afi>	Displays LDP route information for the specified address family and instance. Supported AFI values: ipv4, ipv6, and mpls.
instance <instance-name> ipv4 prefix <ip>	Displays LDP route information for the specified address family of IPv4 prefix and instance.
instance <instance-name> ipv6 prefix <ip>	Displays LDP route information for the specified address family of IPv6 prefix and instance.
instance <instance-name> prefix <ip>	Displays LDP route information for the specified prefix and instance.
instance <instance-name> label <label>	Displays LDP route information for the specified MPLS label and instance.
instance <instance-name> mpls	Displays LDP route information about MPLS labels.

Option	Description
instance <instance-name> mpls label <label>	Displays LDP route information for the specified MPLS label and instance.
label <label>	Displays LDP route information for the specified MPLS label.
ipv4	Displays LDP route information about the IPv4 address family.
ipv4 prefix <ip>	Displays LDP route IPv4 address family information for the specified prefix.
ipv6	Displays LDP route information about the IPv6 address family.
ipv6 prefix <ip>	Displays LDP route IPv6 address family information for the specified prefix.
mpls	Displays LDP route information about MPLS labels.
mpls label <label>	Displays LDP route information for the specified MPLS label.
prefix <ip>	Displays LDP route information for the specified prefix address.

Example: Summary view of LDP Route

```
supervisor@rtbrick>SPINE01: op> show ldp route
Instance: default, AFI: ipv4, SAFI: labeled-unicast
  Prefix/Label         Advertised label  Received label    Next Hop
Interface           Metric
  198.51.100.1/32       3                 -                 -              -
-
  198.51.100.2/32       20065             -                 198.51.100.61
ifl-0/0/0/1      1000000
  198.51.100.3/32       20067             20067             198.51.100.61
ifl-0/0/0/1      2000001
  198.51.100.11/32       3                 -                 -              -
-
  198.51.100.21/32       20065             -                 198.51.100.61
ifl-0/0/0/1      1000000
  198.51.100.31/32       20067             20067             198.51.100.61
ifl-0/0/0/1      2000001
  198.51.100.41/32       3                 -                 -              -
-
  198.51.100.42/32       20065             -                 198.51.100.61
ifl-0/0/0/1      1000000
  198.51.100.43/32       20067             20067             198.51.100.61
ifl-0/0/0/1      2000001
  198.51.100.44/32       3                 -                 -              -
```

```
-
  198.51.100.45/32       20065            -                 198.51.100.61
ifl-0/0/0/1      1000000
  198.51.100.46/32       20067            20067             198.51.100.61
ifl-0/0/0/1      2000001
  198.51.100.47/32       3                -                 -              -
-
  198.51.100.48/32       20065            -                 198.51.100.61
ifl-0/0/0/1      1000000
  198.51.100.49/32       20067            20067             198.51.100.61
ifl-0/0/0/1      2000001
<...>
```

LDP Statistics

Syntax:

show ldp statistics <options>

Option	Description
-	Without any option, the command displays the LDP statistics for all instances.
instance <instance-name>	Displays LDP statistics information about the specified instance.
instance <instance-name> ldp-id <ldp-id>	Displays LDP statistics information about the specified LDP identifier and instance.

Example: LDP statistics information

```
supervisor@rtbrick>SPINE01: op> show ldp statistics
Instance: default, LDP ID: 198.51.100.2:0
  Received messages:
    Initialization: 1, KeepAlive: 558, Notification: 0
    Address: 1, Address Withdraw: 0, Label Mapping: 15
    Label Withdraw: 0, Label Release: 0
  Sent messages:
    Initialization: 1, KeepAlive: 558, Notification: 0
    Address: 1, Address Withdraw: 0, Label Mapping: 15
    Label Withdraw: 0, Label Release: 0
Instance: default, LDP ID: 198.51.100.3:0
  Received messages:
    Initialization: 1, KeepAlive: 557, Notification: 0
    Address: 1, Address Withdraw: 0, Label Mapping: 15
    Label Withdraw: 0, Label Release: 0
  Sent messages:
    Initialization: 1, KeepAlive: 557, Notification: 0
    Address: 1, Address Withdraw: 0, Label Mapping: 15
    Label Withdraw: 0, Label Release: 0
supervisor@rtbrick>SPINE01: op>
```

LDP TCP connection

Syntax:

show ldp tcp connection <options>

Option	Description
-	Without any option, the command displays the TCP connections used by LDP for all instances.
detail	Detailed list view of the TCP connections.
detail instance <instance-name>	Detailed list view of the TCP connections of the specified instance.
instance <instance-name>	TCP connections summary of the specified instance.

Example: Summary view of the LDP TCP connections

```
supervisor@rtbrick>SPINE01: op> show ldp tcp connection
Instance       Local IP             Remote IP            Local port     Remote
port   State
default        198.51.100.1          198.51.100.2                 646
64718   Established
default        198.51.100.1          198.51.100.3                 646
64718   Established
supervisor@rtbrick>SPINE01: op>
```

LDP Clear Commands

Clear commands allow resetting operational states.

Clear LDP Session

Syntax:

clear ldp session <options>

Option	Description
all	Clears all the LDP sessions.
all soft-in	Sends route refresh to all neighbors to receive FEC bindings.

Option	Description
all soft-out	Re-advertises all the routes previously sent to the peers.
instance <instance-name> all	Clears all the LDP sessions for the specified instance.
instance <instance> all soft-in	Sends route refresh to all neighbors to receive FEC bindings for the specified instance.
instance <instance> all soft-out	Re-advertises all the routes previously sent to the peers for the specified instance.
instance <instance-name> peer ldp-id <ldp-id>	Clears the LDP session for the specified instance and peer LDP identifier.
instance <instance> peer ldp-id <ldp-id> soft-in	Sends route refresh to the specific peer to receive FEC bindings for the specified instance and peer ldp-id.
instance <instance> peer ldp-id <ldp-id> soft-out	Re-advertises all the routes previously sent to the specific peer for the specified instance and peer ldp-id.

Example: The example below shows how to clear all the LDP sessions.

```
supervisor@rtbrick>SPINE01: op> clear ldp session all
LDP session cleared with instance default
supervisor@rtbrick>SPINE01: op>
```

Clear LDP Statistics

Syntax:

clear ldp statistics <options>

Option	Description
all	Clears all the LDP statistics.
instance <instance-name> all	Clears all the LDP statistics for the specified instance.
instance <instance-name> peer ldp-id <ldp-id>	Clears the LDP statistics for the specified instance and peer LDP identifier.

Example: The example below shows how to clear all the LDP statistics.

```
supervisor@rtbrick>SPINE01: op> clear ldp statistics all
LDP statistics cleared for instance default
supervisor@rtbrick>SPINE01: op>
```

Clear LDP Neighbor

Syntax:

clear ldp neighbor <options>

Option	Description
all	Clears all the LDP neighbors.
instance <instance-name>	Clears the LDP neighbor for the specified instance.

Example: The example below shows how to clear all the LDP neighbor.

```
supervisor@rtbrick>SPINE01: op> clear ldp neighbor all
LDP neighbor cleared with instance default
supervisor@rtbrick>SPINE01: op>
```

2.6. Policy

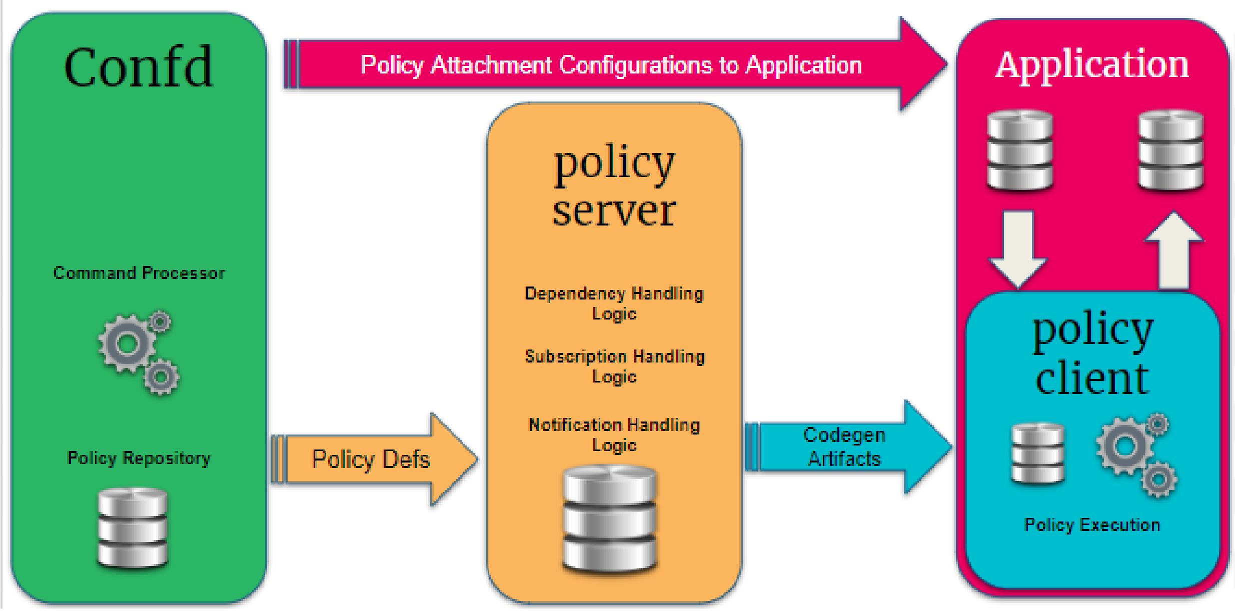
2.6.1. Policy Overview

Policies are rules that allow to control and modify the behavior of routing protocols such as BGP, IS-IS, OSPF, and other supported protocols. The policy framework is generic; it serves multiple purposes and applications. Policies are first created using a common policy configuration and then applied by attaching them to an application like a protocol.

Policy Components

In RtBrick Full Stack, the policy implementation consists of 4 sub-components:

- Policy Repository
- Command Processing Module
- Policy Server, the policy generation and relationship management component
- Policy Client, the policy enforcement component


Confd
Command Processor
Policy Repository
Policy Attachment Configurations to Application
Policy Defs
policy server
Dependency Handling Logic
Subscription Handling Logic
Notification Handling Logic
Codegen Artifacts
Application
policy client
Policy Execution

Policy Repository

The policy repository contains all tables related to policy and the associated list of match criteria.

Command Processing Module

The command processing module is part of the configuration daemon (confd), and it handles user interaction with the policy module. This is the back-end of the Command Line Interface (CLI) and JSON configuration that supports the policy configurations.

This module maps the user-defined configuration into the back-end policy object, which is used by the execution engine (after verification), and it ensures that the policy can be correctly executed. This module relays the user intent via related BDS tables to the policy server.

Policy Server

The policy server is a server component that manages all policy rules in the various policy tables and also code generation of the policies.

The following are the functionalities of the policy server:

- Parses the objects in the policy tables and is an execution engine that generates the code to build the policy rules for evaluation, the relationship between various objects, and relays the intent to the evaluation engine.

- Maintains relationships between various policy constructs such as policy statements, rules, ordinals, and lists.
- Tracks the attachment points so that when policies are modified, the appropriate clients are notified of the relevant new policies.
- Flattens the various relationships and generates a notification table that the clients subscribe to obtain notifications based on specific interest groups.
- Uses dependency table relationships to generate jobs to trigger code generation for various policy components.
- On code generation, the policy server updates a notification table that maintains the mapping between the policy server and the client interest groups. The notification table is a single point for the dissemination of information so that it can generate notifications for clients depending on their subscriptions for policy of interest.
- Policy server notification is generated for the policy clients. A notification is received from the notification table with metadata information that notifies the client if this is a new version of the policy or the original version of the policy. The client uses this information to enforce the policy evaluation and to decide on the version of the policy rule that should be used.

Policy Client

The policy client is a shared library component to which a client daemon, like BGP, IS-IS, OSPF, etc., links. This is the component that performs policy enforcement. It performs the following tasks:

- Links with client daemons like BGP, IS-IS, and OSPF.
- Contains a listener that gets notifications on the availability of a new policy rule that is generated by the policy server.
- Evaluates the compiled rule, and if there are any listeners/ interests, then notifies the components within the client daemon.
- Evaluates any policy configurations on the client daemon and invokes policy processing in response.

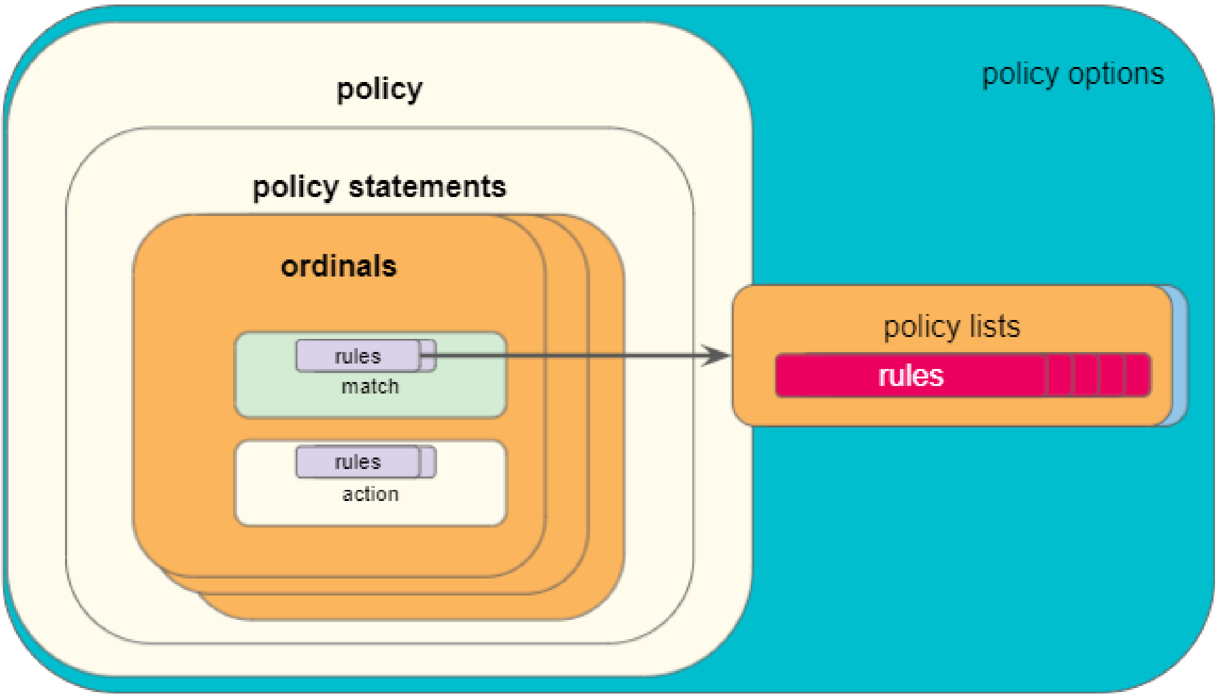
Tables and Subscriptions

The table below shows the various tables and their sharing across various policy components.

Confd	global.policy.list.config global.policy.list.entry.config global.policy.match.rules.config global.policy.statement.config global.policy.ordinal.config global.policy.mapping.list global.policy.mapping.rules	Policy Statement is composed of one or more policy terms. Each term has a match action criteria. In the match and action criteria, either a single element or a list of elements are compared, and actions are taken. The actions include accept, deny, flow-control, etc.
policy.server	global.policy.dependency global.<bds_name>.policy.subscription global.<bds_name>.policy.notification	Policy Server subscribes to all the tables from confd and creates tables that track policy-entry and dependency and notifies clients after code generation.
policy.client	global.<bds_name>.policy.shared.object.cache global.<bds_name>.policy.subscription global.<bds_name>.policy.context	Subscribes to code generation notifications application context and maintains cache of subscribed .so

Policy Building Blocks

The figure below shows the building blocks of a policy.



Statements

A policy is defined by a policy statement. A policy statement is a compound block of policy definition that consists of one or more set of rules called ordinals. A policy statement is exercised in the order defined. The statement name is a globally unique string that is used to identify the policy, and used by the applications.

Ordinals

A policy ordinal is the smallest block to represent a user policy intent and consists of rules for match and action blocks. The match blocks can either define single independent elements like AS path, IP prefix, IP addresses, community, etc., or a list of the elements maintained in a different table.

- An ordinal must be a unique number within the scope of a statement which determines the order of the term execution within a policy statement.
- If no ordinal exist or configured, and if the policy is used, then all routes or objects will be denied.
- A match logic is defined per ordinal. In case of multiple match rules, it defines if all rules (and), or any of the rules (or) have to match.

Match Rules

Match rules define criteria to evaluate and select routes or other objects to which the policy is applied. One or more match rules compose a match block.

- If the match block results in a successful match ("true"), the corresponding action block will executed.
- If the match block result is unsuccessful ("false"), the action block will not be executed, and the next ordinal will be processed.
- If there is no match block, the action block will be executed.

In case of multiple match rules, the behaviour depends on the configured match logic:

- If the match logic is or, the match block result will be succcessful ("true") if any one rule matches. Otherwise, by default it is "false".
- If the match logic is and, the match block result will be successful ("true") if all rules match. Otherwise it is "false".

A match rule can refer to a single discrete value, or a list. A policy list is configured separately and referenced from the policy statement. In case of a list match, the behaviour is as follows:

- If any of the list entries matches the configured value, the match block result will be succcessful ("true"). Otherwise it is "false".
- If the list is defined but empty, the match rule result will be unsuccessful ("false").

Action Rules

Action rules define operations like return-permit or return-deny, or flow control commands like goto-next-ordinal. The action block will be executed if there is a successful match. Otherwise, the next ordinal is processed. The action block can contain one or multiple action rules. In case of multiple action rules, all actions will be applied. The rules are executed in the order of the rule numbers. If there is any termination action, the rules afterwards are not executed anymore.

The action block is optional. The implicit default action is return-deny.

Lists

A policy list is a list of values that can be referenced by a match rule in a policy statement. If you have a number of values like for example route prefixes, it is more efficient to refer to a policy list instead of creating one match rule per prefix.

Policy lists are configured separately and can thereby be maintained more easily. Besides, policy lists can be referenced by multiple policy statements.

Conditions

Policy conditions are configured separately, and can be used as an additional option in policy statements. A policy rule (ordinal) will only be executed if the condition is true. Conditional policies allow to make policy execution depended on certain states of the system, for example:

- Protocol neighbor states
- Presence or absence of a specific route and/or path attribute
- Number of routes in a routing table

One condition is supported per ordinal. A single condition can be attached to multiple policies.

Attachment Points

Policies define set of rules that can be used for various purposes by various applications. Once policies have been created, they need to be applied in order to take effect. Attachment points describe the specific applications and processes to which policies can be applied. RBFS currently supports the following policy attachment points:

- Instance import/export - Policies attached to an instance at the address family level allow to control which routes will be imported into the instance and exported from the instance by BGP. Such import and export policies are commonly used with BGP L3VPNs.
- BGP peer group import/export - Policies attached to BGP peer groups allow to define which routes will be advertised to and accepted from BGP peers. You can attach policies to both instances or peer groups to define the import and export behaviour. You can also combine both attachment points, for example if some policy rules apply generically to the instance and some other rules specifically to a peer or peer group.
- BGP redistribution - You can attach policies to BGP redistribution to define which routes will be redistributed into the BGP process. This is useful if you

would like to redistribute only a sub-set of a type of routes.

- IS-IS redistribution - Policies can be attached to IS-IS redistribution to control which routes will be redistributed into the IS-IS process.
- OSPFv2 redistribution - Policies can be attached to OSPFv2 redistribution to control which routes will be redistributed into the OSPFv2 process.
- IGMP group filtering - You can apply policies to IGMP interface profiles. Such policies act as IGMP group filters when receiving IGMP Membership Report messages.
- IGMP SSM-mapping - Policies are further used to define SSM mapping. SSM mapping policies attached to IGMP interface profiles define how to translate IGMPv2 (*,G) to IGMPv3 (S,G) Reports.

Policy Behaviour

The default behaviour of a policy is deny. This means, if a route or any other object is subject to a policy, by default it will be marked as deny. For example in case of an import policy, it will not be imported. A policy will permit an object, for example import a route, if the route or object successfully matches the match rules, and if there is an action rule with a permit. In addition, further operations like modifying route attributes might be executed.

Policy ordinals are executed in the order of the ordinal numbers.

- If an ordinal results in a terminating action like permit or deny, the policy processing is completed for the respective object. Subsequent ordinals will not be processed.
- If an ordinal does not result in a match, the next ordinal is processed.

There might be situations in which a policy configuration is not complete or not valid. In particular, a policy may contain a match or action type not supported by an attachment point. For example, the ipv4-mcast-group type is not supported by BGP or IS-IS. The following list summarizes the behaviour for such invalid scenarios:

- If a policy is attached but does not exist, all routes or objects will be denied.
- If a policy contains only statements, or only statements and ordinals, but no match and action block, it will deny all.

- If a match or action type is not supported by the attachment point, the policy will ignore the unsupported rule and process only the supported rules. For any ignored rule, the default deny is not impacted. The behaviour will be the same as if the unsupported rule does not exist.

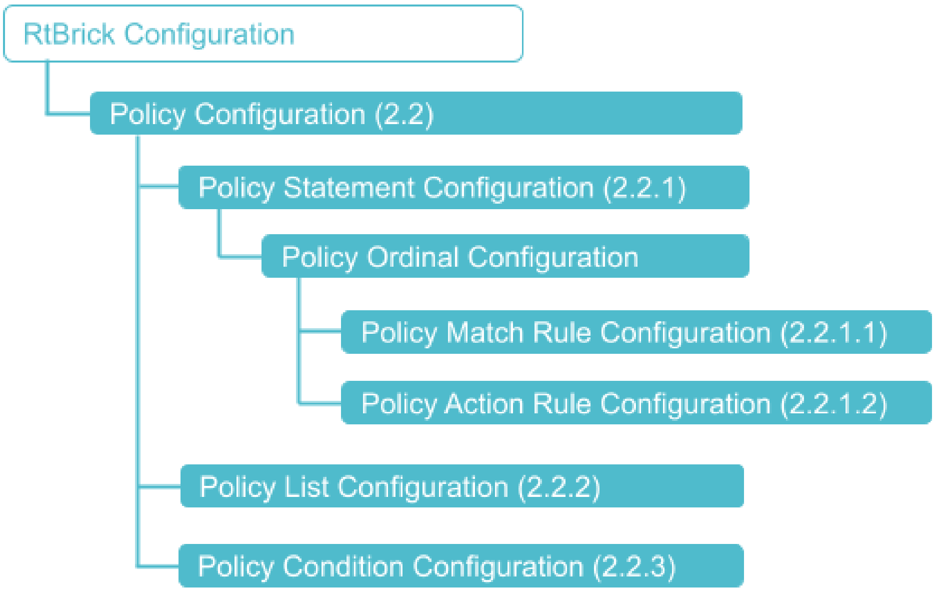
Supported Platforms

Not all features are necessarily supported on each hardware platform. Please refer to the RBFS Platform Guide for the features and the sub-features supported or not supported for each platform.

2.6.2. Policy Configuration

Configuration Hierarchy

The diagram illustrates the policy configuration hierarchy.


RtBrick Configuration
Policy Configuration (2.2)
Policy Statement Configuration (2.2.1)
Policy Ordinal Configuration
Policy Match Rule Configuration (2.2.1.1)
Policy Action Rule Configuration (2.2.1.2)
Policy List Configuration (2.2.2)
Policy Condition Configuration (2.2.3)

Configuration Syntax and Commands

The following sections describe the policy configuration syntax and commands.

Configuring Policy Statements

Syntax:

set policy statement <policy-name> **ordinal** <number> <attribute> <value>

Attribute	Description
<policy-name>	Name of the policy statement. Policy names can contain alphanumeric characters and underscore character. They must not include special characters like hyphen. For example, **BGP-EXPORT** is not supported, whereas **BGP_EXPORT** is supported. A valid name cannot start with a number but it can contain numbers and underscore (_) in the string. The length of the names should not exceed 64 characters.
<number>	Specifies the ordinal number.
description <text>	Description of the ordinal.
match <…>	Match configuration hierarchy. Please refer to section 2.2.1.1 for the match rule configuration.
match-logic <value>	Specifies the match logic. Supported values are and and or.
action <…>	Action configuration hierarchy. Please refer to section 2.2.1.2 for the action rule configuration.
condition <condition-name>	Optionally, apply a policy condition. Please refer to section 2.2.3 for the policy condition configuration.

Configuring Match Rules

Syntax:

set policy statement <policy-name> **ordinal** <number> **match rule** <number> <attribute> <value>

Attribute	Description
rule <number>	Specifies the match rule number.
type <attribute-type>	Specifies the attribute type. Please refer to section 2.2.1.1.1 for supported attribute types.
match-type <match-type>	Specifies the match type. Please refer to section 2.2.1.1.1 for supported match types per attribute, and to section 2.2.1.1.2 for descriptions of the match types.

Attribute	Description
value <value>	Attribute value. This is the actual value of the attribute to match, for example an IP prefix, a metric, or a community.
value-type <value-type>	Attribute value type. Supported types are discrete, this is a single value, and list, a list of values defined in a policy list.

Attribute and Match Types

Attribute Type	Match Types Supported
ipv4-prefix	regex exact longer or-longer prefix-length-exact prefix-length-greater prefix-length-greater-or-exact
ipv6-prefix	regex exact longer or-longer prefix-length-exact prefix-length-greater prefix-length-greater-or-exact
route-distinguisher	regex exact
community	regex exact exists
extended-community	regex exact exists
large-community	regex exact exists

Attribute Type	Match Types Supported
as-path	regex exact exists
cluster-list	regex exact exists
source	regex exact
sub-source	regex exact
originator-identifier	regex exact
peer-router-id	regex exact
ipv4-nexthop	regex exact
ipv6-nexthop	regex exact
label	regex exact
peer-ipv4	regex exact
peer-ipv6	regex exact
sid	regex exact
sid-flag	regex exact
external	exact

Attribute Type	Match Types Supported
igp-metric	regex exact greater greater-or-exact less less-or-exact

Match Types

Match Types	Description
regex	An attribute can be matched using a standard Linux egrep regular expression. Example: "label": "label-op:push,label:206,bos:1" In this example, the label is a 64bit number, which has label value, bos, and operation encoding. A regex is used to match the string which is displayed in the table dump, that is, label-op:push,label:206,bos:1 not the 64bit value. The same is applicable to an array type attribute. A regex can be written to the string which is visible in the table dump output.
exact	Value configured in the command must be same as application attribute value
exists	This is applicable only for array type attribute; an exist match is the one where value configured in the command must exist in the application attribute value which is an array.
less	The application attribute value must be less than the value configured in the command
less-or-exact	The application attribute value must be less than or exact value configured in the command
greater	The application attribute value must be greater than the value configured in the command

Match Types	Description
greater-or-exact	The application attribute value must be greater than or exact value configured in the command
greater-longer	The route shares the same most-significant bits (described by prefix-length), and prefix-length is greater than the route’s prefix length
greater-or-longer	The route shares the same most-significant bits (described by prefix-length), and prefix-length is equal to or greater than the route’s prefix length.
longer	The route address shares the same most-significant bits as the match prefix (destination-prefix or source-prefix). The number of significant bits is described by the prefix-length component of the match prefix. The match will be performed only for prefixes longer than the one supplied for matching.
or-longer	The route address shares the same most-significant bits as the match prefix (destination-prefix or the source-prefix). The number of significant bits is described by the prefix-length component of the match prefix. The match is performed for the prefix supplied and for any prefixes that are subnets of it.
prefix-length-exact	The application attribute value whose prefix length must be lesser than or exact the value configured in the command
prefix-length-greater	The application attribute value whose prefix length must be greater than the value configured in the command
prefix-length-greater-or-exact	The application attribute value whose prefix length must be greater than or exact value configured in the command

Configuring Action Rules

Syntax:

set policy statement <policy-name> **ordinal** <number> **action rule** <number> <attribute> <value>

Attribute	Description
rule <number>	Specifies the action rule number.

Attribute	Description
operation <operation-type>	Specifies the operation type. Please refer to section 2.2.1.2.1 for supported operations, and to section 2.2.1.2.2 for operations per attribute.
type <attribute-type>	Specifies the attribute type. Please refer to section 2.2.1.2.2 for supported attribute types.
value <value>	Specifies the operation value.

Operation Types

Operation Type	Description
add	The application attribute value will be added with the value configured in the command
append	The application attribute value will be appended with the value configured in the command
delete-attribute	Deletes the attribute from the route/BDS object, that is, clearing all the info for that specific attribute in the object
divide	The application attribute value will be divided with the value configured in the command
goto-next-ordinal	If next term exists, then next term is executed and the policy result is decided based on the result of the execution
multiply	The application attribute value will be multiplied with the value configured in the command
overwrite	The application attribute value will be overwritten with the value configured in the command
prepend	The application attribute value will be prefixed with the value configured in the command
return-deny	Stops policy execution and returns result as deny (resulting route/BDS object to be denied)
return-permit	Stops policy execution and return result as permit (resulting route/BDS object to be permitted)

Operation Type	Description
subtract	The application attribute value will be subtracted with the value configured in the command. If the result of the subtraction results in a number less than 0, the value "0" is used.

Attribute Types and Supported Operations

Attribute Type	Operation Types Supported
ipv4-prefix	overwrite
ipv6-prefix	overwrite
route-distinguisher	overwrite
community	append prepend overwrite
extended-community	append prepend overwrite
large-community	append prepend overwrite
as-path	append prepend overwrite
cluster-list	append prepend overwrite
source	overwrite
sub-source	overwrite
originator-identifier	overwrite
peer-router-id	overwrite
ipv4-nexthop	overwrite
ipv6-nexthop	overwrite

Attribute Type	Operation Types Supported
label	overwrite
peer-ipv4	overwrite
peer-ipv6	overwrite
sid	overwrite
sid-flag	overwrite
external	overwrite
igp-metric	add subtract multiply divide overwrite
metric-type	overwrite

Example: Policy statement configuration

```
{
    "rtbrick-config:policy": {
      "statement": [
        {
          "name": "EXPORT_POLICY1",
          "ordinal": [
            {
              "ordinal": 10,
              "description": "Add BGP community to direct routes",
              "match-logic": "and",
              "match": {
                "rule": [
                  {
                    "rule": 1,
                    "type": "ipv6-prefix",
                    "value-type": "discrete",
                    "match-type": "or-longer",
                    "value": "2001:db8:0:60::/32"
                  },
                  {
                    "rule": 2,
                    "type": "source",
                    "value-type": "discrete",
                    "match-type": "exact",
                    "value": "direct"
                  }
                ]
              },
              "action": {
                "rule": [
                  {
```

```
                  "rule": 1,
                  "type": "community",
                  "operation": "append",
                  "value": "100:1"
                },
                {
                  "rule": 2,
                  "operation": "return-permit"
                }
              ]
            }
          },
          {
            "ordinal": 20,
            "description": "Allow any other route",
            "action": {
              "rule": [
                {
                  "rule": 1,
                  "operation": "return-permit"
                }
              ]
            }
          }
        ]
      }
    ]
  }
}
```

Configuring Policy Lists

Syntax:

set policy list <list-name> <list-type> **ordinal** <ordinal-number> **value** <value>

Attribute	Description
<list-name>	Name of the policy list

Attribute	Description
<list-type>	Type of the policy list. The following types of lists are supported: * as-path * cluster-list * community * route-distinguisher * extended-community * ipv4-address * ipv4-mcast-group * ipv4-prefix * ipv6-address * ipv6-prefix * large-community * mac-address * mpls-label * source * sub-source
<ordinal-number>	The number of the list entry.
<value>	The value of the list entry, for example an IP prefix, or a community.

Example: Policy list configuration

```
{
    "rtbrick-config:policy": {
      "list": [
        {
          "name": "PREFIX_LIST1",
          "type": "ipv6-prefix",
          "ordinal": [
            {
              "ordinal": 1,
              "value": "2001:db8:0:10::/32"
            },
            {
              "ordinal": 2,
              "value": "2001:db8:0:25::/32"
            },
            {
              "ordinal": 3,
              "value": "2001:db8:0:30::/32"
            }
          ]
        }
      ]
    }
}
```

Configuring Policy Conditions

Policy conditions refer to certain states of the system represented in BDS tables. You need to specify the table, the daemon (BD) that needs to resolve the condition, and if applicable, the attributes used to define the condition.

There are two types of conditions:

- Match on certain attributes in BDS table objects.
- Match on the number of objects in a BDS table.

Configuring Table Match

Syntax:

set policy condition <condition-name> **table** <attribute> <value>

Attribute	Description
<condition-name>	Name of the policy condition
name <table-name>	Name of the BDS table
bd <bd-name>	Name of the BD which resolves the condition. Currently supported BDs are: ifmd, ribd, mribd, pppoed, subscriberd, ipoed, l2tpd, pimd, igmp.iod, isis.iod, ospf.appd ospf.iod
count <number>	Optionally, match on the number of objects in a table
match <type>	Type of match for an object count. Supported match types are: equal greater greater-or-equal less less-or-equal

Configuring Attribute Match

You can configure attributes to define a condition. The attributes refer to objects in the table configured in the section above. Please note:

- You can match on multiple attributes.
- In order to identify matching objects, you need to specify all attributes which are primary keys in the table.
- Attribute configuration is not required for conditions matching on the number of table objects using the count option.

Syntax:

set policy condition <condition-name> **attribute** <name> <attribute> <value>

Attribute	Description
<condition-name>	Name of the policy condition
attribute <name>	Name of the attribute in the BDS table
value <value>	Value of the attribute to match
match <type>	Type of the match. Supported match types are: equal exists greater greater-or-equal less less-or-equal regex

Example: Policy condition configuration

```
{
    "rtbrick-config:policy": {
      "condition": [
        {
          "condition_name": "precheck",
          "table": {
            "name": "global.instance",
            "bd": "bgp.iod.1"
          },
          "attribute": [
            {
              "name": "instance_name",
              "match": "equal",
              "value": "default"
            }
          ]
        }
      ]
    }
  }
```

Attaching Policies

Once a policy has been created, they need to be applied to an application like a routing protocol to take effect.

BGP Attachment Points

- Instance import/export
- BGP peer group import/export
- BGP redistribution

For attaching policies to the BGP protocol, please refer to the RBFS BGP User Guide.

IS-IS Attachment Points

- IS-IS redistribution

OSPFv2 Attachment Points

- OSPFv2 redistribution

For attaching policies to the IS-IS protocol, please refer to the RBFS IS-IS User Guide.

IGMP Attachment Points

- IGMP group filtering
- IGMP SSM-mapping

For attaching policies to the IGMP protocol, please refer to the RBFS IP Multicast Routing Configuration Guide.

Sample Configurations

Example 1: BGP export policy referencing a policy list

```
supervisor@leaf1: cfg> show config policy
{
  "rtbrick-config:policy": {
    "list": [
      {
        "name": "PREFIX_LIST2",
        "type": "ipv6-prefix",
        "ordinal": [
          {
            "ordinal": 1,
            "value": "2001:db8:0:60::/32"
          },
          {
            "ordinal": 2,
            "value": "2001:db8:0:80::/64"
          },
          {
            "ordinal": 3,
            "value": "2001:db8:0:110::/64 "
          }
        ]
      }
    ],
    "statement": [
      {
        "name": "EXPORT_POLICY2",
        "ordinal": [
          {
            "ordinal": 10,
            "description": "Add community to direct routes",
            "match": {
              "rule": [
                {
                  "rule": 1,
                  "type": "source",
                  "value-type": "discrete",
                  "match-type": "exact",
                  "value": "direct"
                }
              ]
            },
            "action": {
              "rule": [
                {
                  "rule": 1,
                  "type": "community",
                  "operation": "append",
                  "value": "100:1"
                },
                {
                  "rule": 2,
                  "operation": "return-permit"
                }
              ]
            }
          },
          {
            "ordinal": 20,
            "description": "Allow list of routes",
            "match": {
              "rule": [
                {
                  "rule": 1,
                  "type": "ipv6-prefix",
                  "value-type": "list",
                  "match-type": "or-longer",
                  "value": "PREFIX_LIST2"
                }
              ]
            },
            "action": {
              "rule": [
                {
                  "rule": 1,
                  "operation": "return-permit"
                }
              ]
            }
          },
          {
            "ordinal": 30,
            "description": "Deny any other route",
            "action": {
              "rule": [
                {
                  "rule": 1,
                  "operation": "return-deny"
                }
              ]
            }
          }
        ]
      }
    ]
  }
}

supervisor@leaf1: cfg> show config instance default protocol bgp peer-group spine
address-family ipv6 unicast
{
  "rtbrick-config:address-family": [
    {
      "afi": "ipv6",
      "safi": "unicast",
      "policy": {
        "export": "EXPORT_POLICY2"
      }
    }
  ]
}
```

Example 2: IGMP filter policy

```
supervisor@leaf1: cfg> show config policy
```

```
{
  "rtbrick-config:policy": {
    "statement": [
      {
        "name": "IGMP_FILTER",
        "ordinal": [
          {
            "ordinal": 1,
            "description": "IGMP group filter",
            "match-logic": "or",
            "match": {
              "rule": [
                {
                  "rule": 1,
                  "type": "ipv4-mcast-group",
                  "value-type": "discrete",
                  "match-type": "or-longer",
                  "value": "198.51.100.30/24"
                },
                {
                  "rule": 2,
                  "type": "ipv4-mcast-group",
                  "value-type": "discrete",
                  "match-type": "or-longer",
                  "value": "198.51.100.40/24"
                }
              ]
            },
            "action": {
              "rule": [
                {
                  "rule": 1,
                  "operation": "return-permit"
                }
              ]
            }
          }
        ]
      }
    ]
  }
}

supervisor@leaf1: cfg> show config multicast-options igmp
{
  "rtbrick-config:igmp": {
    "interface-profile": [
      {
        "profile-name": "PROFILE1",
        "filter-policy": "IGMP_FILTER"
      }
    ]
  }
}
```

2.6.3. Policy Operational Commands

Policy Test

You can use the policy test feature to test a policy before attaching it to a protocol or an instance.

Perform the following tasks:

- Step 1: Identify the brick daemon that will process the policy and the table to which the policy will be applied.
- Step 2: Execute the 'test policy run' command.

Example: Testing a BGP VPN export policy

```
supervisor@leaf1: op> test policy run bgp.appd.1 policy-name VPN_V4_EXPORT table
default.bgp.rib-in.import.ipv4.vpn-unicast
```

- Step 3: View the test results.

The policy test feature will create two result tables. The result table ending with ".policy.permit" will show all objects permitted by the policy, the one ending with ".policy.deny" will show all objects denied by the policy.

Example: Viewing the result tables

```
supervisor@leaf1: op> show datastore bgp.appd.1 table default.bgp.rib-
in.import.ipv4.vpn-unicast.policy.permit
<...>
supervisor@leaf1: op> show datastore bgp.appd.1 table default.bgp.rib-
in.import.ipv4.vpn-unicast.policy.deny
<...>
```

- Step 4: Clear the result tables

You can clear the result tables using the 'test policy clear' command. Apply the clear command to the same table for which you have run the policy test.

Example: Clearing the result tables

```
supervisor@leaf1: op> test policy clear bgp.appd.1 policy-name VPN_V4_EXPORT table
default.bgp.rib-in.import.ipv4.vpn-unicast
```

2.7. Static Routing

2.7.1. Static Routing Overview

Static routing allows a network administrator to configure routes manually. Using the RtBrick CLI, you can configure static IPv4, IPv6, MPLS, and multicast routes.

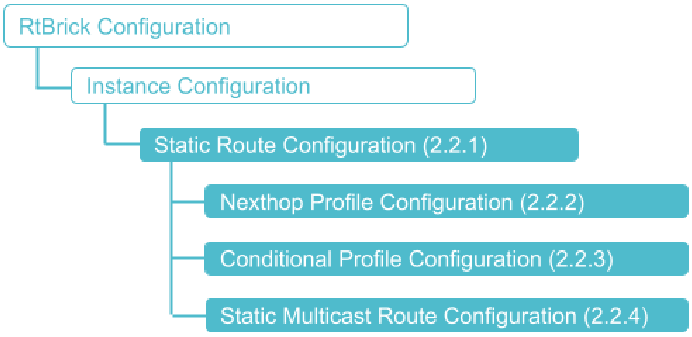
Supported Platforms

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

2.7.2. Static Routing Configuration

Configuration Hierarchy

The diagram illustrates the static routes configuration hierarchy.


RtBrick Configuration
Instance Configuration
Static Route Configuration (2.2.1)
Nexthop Profile Configuration (2.2.2)
Conditional Profile Configuration (2.2.3)
Static Multicast Route Configuration (2.2.4)

Configuration Syntax and Commands

The following sections describe the static route configuration syntax and commands. In RBFS, next hops of static routes are configured separately, and referenced by the actual routes.

Static Route Configuration

This section describes how to configure the static route itself.

Syntax

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

Attribute	Description
<instance-name>	Name of the routing instance
<afi> <prefix | label> (true | false)	Supported AFIs are ipv4, ipv6, and mpls. In case of IPv4 or IPv6, configure the prefix. In case of MPLS, configure the incoming label and BOS flag.
<safi>	Supported SAFIs are unicast, labeled-unicast, and multicast.
<nexthop-profile>	Name of the nexthop profile

Example: Static Route Configuration

```
{
    "rtbrick-config:route": {
      "ipv4": [
        {
          "prefix4": "198.51.100.15/24",
          "safi": "unicast",
          "nexthop-profile": "nexthop1",
          "preference": 20

        }
      ],
      "ipv6": [
        {
          "prefix6": "2001:db8:0:117::/32",
          "safi": "unicast",
          "nexthop-profile": "nexthop2"
        }
      ],
      "mpls": [
        {
          "in-label": 8888,
          "in-bos": "true",
          "safi": "unicast",
          "nexthop-profile": "nexthop1"
        }
      ]
    }
  }
```

Nexthop Profile Configuration

You can group various nexthop parameters with a nexthop profile name and instance, and associate this nexthop profile with multiple routes.

Syntax

set instance <instance-name> **static nexthop-profile** <name> <attribute> <value>

Attribute	Description
<instance-name>	Name of the routing instance
nexthop-profile <name>	Nexthop profile name
exit-interface <exit-interface>	Exit interface name
lookup-afi (ipv4 | ipv6 | mpls)	Lookup routing table address family where the nexthop will be resolved.
lookup-instance <lookup-instance>	Lookup routing table instance where the nexthop will be resolved.
lookup-safi (labeled-unicast | multicast | unicast)	Lookup routing table subsequent address family where the nexthop will be resolved.
nexthop <address>	IPv4/IPv6 nexthop address
out-bos (true | false)	Label BOS
out-label <out-label>	Label to be pushed
resolve-direct true	The option restricts all routes from resolving the nexthop of a static route instead, it allows only the direct routes to resolve the nexthop of a static route.

Example: Nexthop Profile Configuration

```
{
    "rtbrick-config:static": {
      "nexthop-profile": [
        {
          "name": "nexthop1",
          "nexthop": "198.51.100.145",
          "out-label": 4444
        },
        {
          "name": "nexthop3",
          "exit-interface": "ifp-0/0/4/4"
        }
      ]
    }
  }
```

- If you do not provide lookup-instance, lookup-afi and lookup-

> safi values, default values will be used to install the route.
>
> - The exit interface attribute is mandatory for link-local nexthop.

Conditional Profile Configuration

By using the conditional static route feature, you can make specific routes conditional. These conditional routes are installed only if the specified condition is satisfied.

You can group various conditional parameters such as match-instance, match-afi, match-safi, compare-operation, compare-type, and compare-value with a conditional profile name, and associate this conditional profile with multiple routes.

Syntax:

set instance <instance-name> **static conditional-profile** <name> <attribute> <value>

Attribute	Description
conditional <name>	Conditional profile name
compare-operation greater-then	Conditional routing compare operation
compare-type route-count	Conditional routing compare type
compare-value <compare-value>	Conditional routing condition value
match-instance <instance-name>	Routing instance where the condition will be checked.
match-afi (ipv4 |ipv6 |mpls)	Routing tables address family (AFI) for which the condition will be checked.
match-safi (labeled-unicast |multicast |unicast)	Routing table subsequent address family (SAFI) for which the condition will be checked.

Example: Conditional Profile Configuration

```
{
    "rtbrick-config:conditional-profile": [
      {
        "name": "c2",
        "match-instance": "default",
        "match-afi": "ipv4",
        "match-safi": "unicast",
        "compare-type": "route-count",
        "compare-operation": "greater-than",
        "compare-value": 20
      }
    ]
  }
```

Static Multicast Route Configuration

Syntax:

set instance <instance-name> **static route multicast4** <attribute> <value

Attribute	Description
<instance-name>	Name of the routing instance
<source>	IPv4 multicast source address
<group>	IPv4 multicast group address

Example: Static Multicast Route Configuration

```
{
    "rtbrick-config:static": {
      "route": {
        "multicast4": [
          {
            "source": "198.51.100.15/24",
            "group": "198.51.100.35/24",
            "nexthop-profile": "nexthop3"
          }
        ]
      }
    }
}
```

2.7.3. Static Routing Operational Commands

Show Commands

Static Routes Created by staticd

These commands show static routes as created by the static route daemon (staticd).

Syntax:

show static route <options>

Attribute	Description
<afi>	Supported AFIs are ipv4, ipv6, and mpls.
<safi>	Supported SAFIs are unicast, labeled-unicast, and multicast.
instance <name>	Static routes for an instance

Example: List static routes for all instances

```
supervisor@dev1: cfg> show static route
Instance: default, AFI: ipv4, SAFI: unicast
Prefix/Label              Pref    Next Hop                  Interface
198.51.100.100/24          2       198.51.100.22                   -
Instance: default, AFI: ipv6, SAFI: unicast
Prefix/Label              Pref    Next Hop                  Interface
2001:db8:0:334::/32                2     2001:db8:0:99::                      -
```

Static Routes in the Routing Table

These commands show the static routes included in the final routing table.

Syntax:

show route <options> **source static** <options>

Attribute	Description
<afi>	Supported AFIs are ipv4, ipv6, and mpls.
<safi>	Supported SAFIs are unicast, labeled-unicast, and multicast.
detail	Detailed route information
instance <name>	Routing table information for a specific instance
label <value>	Destination label
mpls	Address family
prefix <value>	Destination prefix

Attribute	Description
source	Source of the routing information

Example 1: List static routes information

```
supervisor@dev1: cfg> show route ipv4 source static
Instance: default, AFI: ipv4, SAFI: unicast
Prefix/Label                             Source            Pref    Next Hop
Interface
198.51.100.100/24                         static            2       198.51.100.22
ifl-0/0/1/4
supervisor@dev1: cfg>
```

Example 2: List detailed static routes information

```
supervisor@dev1: cfg> show route ipv4 source static detail
Instance: default, AFI: ipv4, SAFI: unicast
198.51.100.100/24
  Source: static, Preference: 2
    Next Hop: 198.51.100.22
      Covering prefix: 198.51.100.22/24
      Next Hop type: direct, Next Hop action: None
      Resolved in: default-ipv4-unicast
      Egress interface: ifl-0/0/1/4, NextHop MAC: 7a:00:81:64:04:04
```

Example 3: List MPLS route information

```
supervisor@rtbrick: cfg> show route mpls
Instance: default, AFI: mpls, SAFI: unicast
Prefix/Label    Source            Pref    Next Hop                Interface
20010           bgp               170     2001:db8:0:110::        ifl-0/0/17/1001
20011           bgp               170     2001:db8:0:110::        ifl-0/0/17/1001
20012           bgp               170     2001:db8:0:110::        ifl-0/0/17/1001
20013           bgp               170     2001:db8:0:110::        ifl-0/0/17/1001
20014           bgp               170     2001:db8:0:110::        ifl-0/0/17/1001
20015           bgp               170     2001:db8:0:110::        ifl-0/0/17/1001
20016           bgp               170     2001:db8:0:110::        ifl-0/0/17/1001
20017           bgp               170     2001:db8:0:110::        ifl-0/0/17/1001
20018           bgp               170     2001:db8:0:110::        ifl-0/0/17/1001
```

3. Layer 2 Services

3.1. L2X

3.1.1. L2X Overview

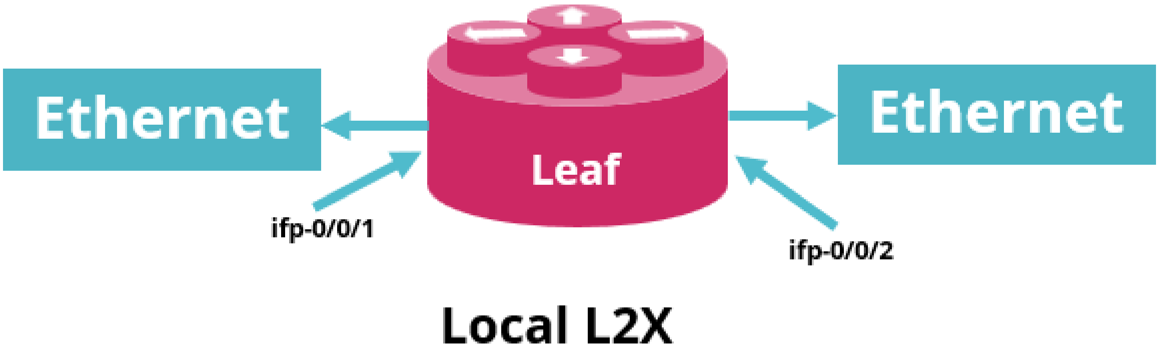
Layer 2 Cross-Connect (L2X) is a data plane feature that connects two physical ports (IFPs) using Layer 2 switching. L2X can switch the traffic between two IFPs to provide the trunk service for an Ethernet switch.

Local and Remote L2X

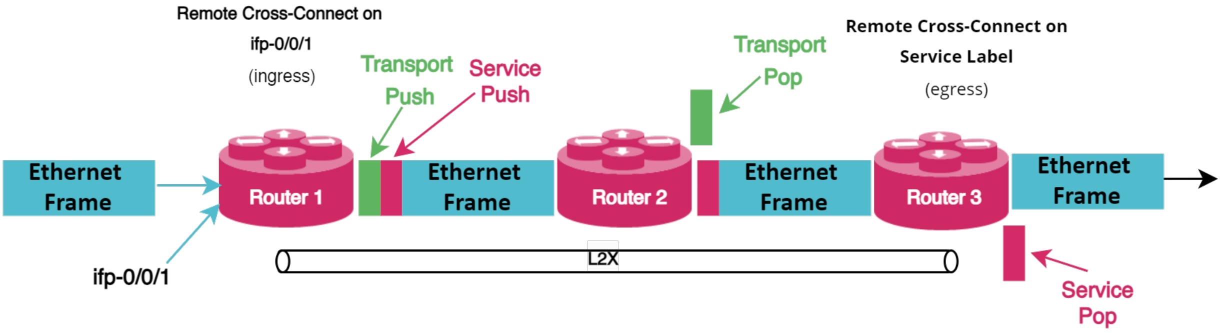
Local L2X refers to an L2 connection between two ports or VLANs on the same device. In a local L2X, both interfaces are on the same router. The L2X can switch Layer 2 (frame) traffic between the ports. Based on the configuration, these cross connects can be uni-directional as well as bi-directional.

Remote L2X refers to L2 connection between two ports located on two different devices. In a remote L2X, the interfaces are located on two different routers and it requires an MPLS tunnel to transport the traffic between the two routers.

Local L2X: The following figure shows the Local L2X scenario.



Remote L2X: The following figure shows the remote L2X scenario.



Unidirectional and Bidirectional L2X

Unidirectional refers to data either sent or received in one direction and Bidirectional implies the flow of traffic between two routers in both directions.

The bidirectional cross connect feature helps you to establish cross connection between two local ports with an L2X configuration. Bi-directional attribute is applicable only to local cross connect. Bidirectional connectivity requires a pair of unidirectional L2X or a single bidirectional L2X.

> The VLAN operations are not supported for bi-directional local cross-connect.

Ingress and Egress in L2X

In L2X, ingress traffic is incoming traffic that enters the boundary of a network and egress traffic implies outgoing traffic that exits an entity or a network boundary.

Port and VLAN Cross-connects

Both port and VLAN cross-connects switches Layer 2 traffic from input interface to output interface. A port cross-connect switches all Layer 2 traffic arriving at an input interface, but a VLAN cross-connect only switches the Layer 2 traffic associated with a specific VLAN. A port-based L2X indicates a port-only configuration, so there are no VLANs involved.

Both single-tag and double-tagged (inner and outer VLAN tags) are supported. The port and VLAN L2X support both local and remote L2X configurations. In remote L2X connections, the VLAN cross-connects are typically configured on the MPLS tunnel ingress router.

Untagged traffic on L2X interfaces is also supported. However, there is no way to select only untagged traffic for cross-connecting. Therefore, only port cross connects are supported for untagged traffic.

L2X 802.1ad Ethertype Support

RBFS supports VLAN operations such as VLAN add, VLAN swap, and VLAN delete on egress interface. RBFS supports similar functionality at the ingress side as well. That is, RBFS supports the following VLAN operations:

- Single-VLAN-Add with an option to configure encapsulation (that is, 802.1q or 802.1ad)
- Single-VLAN-Delete
- Swap-Outer-VLAN

By default the encapsulation method is 802.1q. If an encapsulation method is not specified, 802.1q is the default mode.

In addition to setting the Ethertype for a VLAN operation, the 802.1ad support includes that ingress traffic for all tagged match options will match on both Ethertype 0x8100 (802.1q) and 0x88a8 (802.1ad) by default.

VLAN Operations

RBFS supports VLAN operations such as VLAN add, VLAN swap and VLAN delete on Ingress and Egress interfaces.

The current functionality has been extended to all the existing CLIs to accept ingress and egress VLAN operations and Ingress and Egress VLAN encapsulation values.

Both 802.1q and 802.1ad encapsulations are supported. The default encapsulation is 802.1q.

Traffic will be matched at ingress direction based on the match criterion. RtBrick Full Stack (RBFS) supports the following match parameters.

On a physical interface, there are five different match types. Traffic can be matched based on the following:

1. (ifp)

2. (ifp, outer_vlan)
3. (ifp, outer_vlan, inner_vlan)
4. (ifp, outer_vlan, any inner_vlan)
5. (ifp, any vlan)

Some of the match types are mutually exclusive. For example, (ifp, outer_vlan, inner_vlan) and (ifp, outer_vlan, any inner_vlan) configuration on the same interface throws errors.

If ifp, any vlan match type is configured with any other match type, it will create conflicts.

The match-type attribute is mandatory for match-untagged, match-any and match-inner-any match criteria.

Supported Match Type Validations

The following table shows the supported match type validations.

The asterisk * indicates *any* or *no vlan* tags.

Cases	Configuration A	Configuration B	Support
Case 1 : IFP A, *	IFP A,*	IFP A, ov 10	No
	IFP A,*	IFP A, ov 10, iv 20	No
	IFP A,*	IFP A, ov 10, *	No
	IFP A,*	IFP A, untagged	No
Case 2: IFP A, untagged	IFP A, untagged	IFP A, *	No
	IFP A, untagged	IFP A, ov 10	Yes
	IFP A, untagged	IFP A, ov 30, iv 20	Yes
	IFP A, untagged	IFP A, ov 20, *	Yes

Cases	Configuration A	Configuration B	Support
Case 3: IFP A, outer_vlan:	IFP A, ov 10	IFP A, *	No
	IFP A, ov 10	IFP A, ov 10, *	No
	IFP A, ov 10	IFP A, ov 20	Yes
	IFP A, ov 10	IFP A, ov 10 , iv 20	No
	IFP A, ov 10	IFP A, ov 40 , iv 7	Yes
	IFP A, ov 10	IFP A, ov 30, *	Yes
	IFP A, ov 10	IFP A, untagged	Yes
Case 4: IFP A, outer_vlan, inner_vlan:	IFP A, ov 10, iv 20	IFP A, *	No
	IFP A, ov 10, iv 20	IFP A, ov 10, *	No
	IFP A, ov 10, iv 20	IFP A, ov 10	No
	IFP A, ov 10, iv 20	IFP A, ov 30	Yes
	IFP A, ov 10, iv 20	FP A, ov 20 , *	Yes
	IFP A, ov 10, iv 20	IFP A, untagged	Yes
	IFP A, ov 10, iv 20	IFP A, ov 10, iv 30	Yes
Case 5: IFP A, outer_vlan, *	IFP A, ov 10, *	IFP A, *	No
	IFP A, ov 10, *	IFP A, ov 10	No
	IFP A, ov 10, *	IFP A, ov 10, iv 20	No
	IFP A, ov 10, *	IFP A, ov 20, iv 7	Yes
	IFP A, ov 10, *	IFP A, ov 30	Yes
	IFP A, ov 10, *	IFP A, untagged	Yes
	IFP A, ov 10, *	IFP A, ov 40, *	Yes

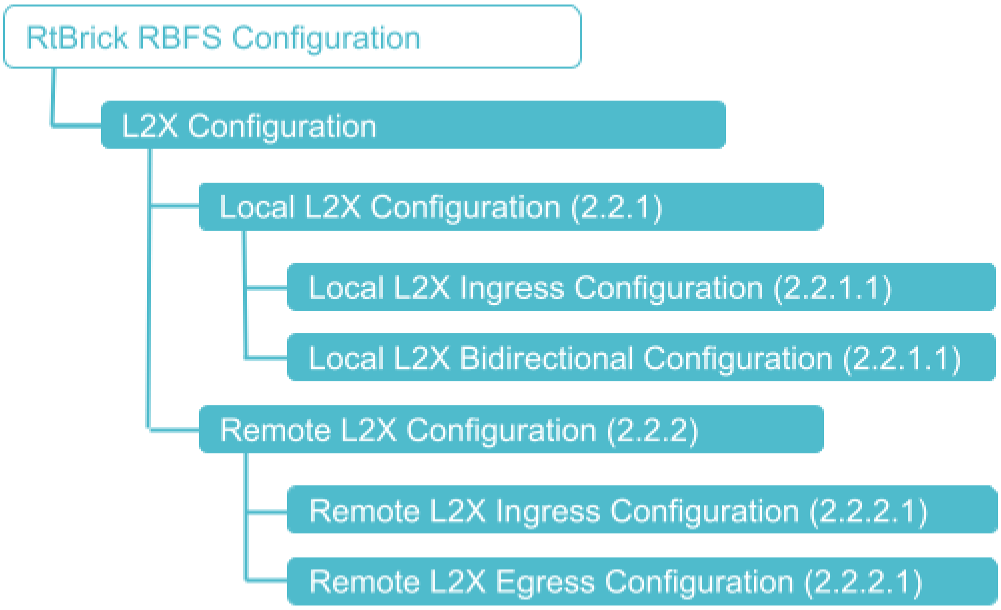
Supported Platforms

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

3.1.2. L2X Configuration

Configuration Hierarchy

The diagram illustrates the L2X configuration hierarchy.


RtBrick RBFS Configuration
L2X Configuration
Local L2X Configuration (2.2.1)
Local L2X Ingress Configuration (2.2.1.1)
Local L2X Bidirectional Configuration (2.2.1.1)
Remote L2X Configuration (2.2.2)
Remote L2X Ingress Configuration (2.2.2.1)
Remote L2X Egress Configuration (2.2.2.1)

Configuration Syntax and Commands

The following sections describe the L2X configuration syntax and commands.

Local L2X Configuration

The following sections describe the Local L2X unidirectional and bidirectional configurations.

Local L2X Ingress Configuration

This configuration enables local unidirectional L2X (Local Cross-Connect) on the same device.

Syntax:

set l2x name <l2x-name> **ingress** <attribute> <value>

Attribute	Description
<l2x-name>	Name of L2X
description <description>	(Optional) L2X description

Attribute	Description
egress-vlan-encapsulation <encapsulation>	(Optional) Egress VLAN encapsulation value
egress-vlan-operation <vlan-action>	(Optional) Outgoing VLAN operation
incoming-inner-vlan <vlan-id>	(Optional) Incoming inner VLAN
incoming-interface <incoming-interface>	(Mandatory) Incoming physical interface name
incoming-outer-vlan <vlan-id>	(Optional) Incoming outer VLAN
ingress-outer-vlan <vlan-id>	(Optional) Outer VLAN at ingress side
ingress-vlan-encapsulation <encapsulation>	(Optional) Ingress VLAN encapsulation value
ingress-vlan-operation <vlan-action>	(Optional) VLAN operation on ingress side outer VLAN
match-type <match-type>	(Mandatory) L2X match type
outgoing-interface <outgoing-interface>	(Mandatory) Outgoing physical interface name
outgoing-outer-vlan <vlan-id>	(Optional) Outgoing outer VLAN

Example 1: Local L2X Ingress Configuration with Port Match

```
{
    "rtbrick-config:l2x": {
      "name": [
        {
          "name": "test1",
          "direction": "ingress",
          "incoming-interface": "ifp-0/1/64",
          "outgoing-interface": "ifp-0/1/66",
          "match-type": "match-any"
        }
      ]
    }
  }
```

Example 2: Local L2X Ingress Configuration with VLAN Match

```
{
    "rtbrick-config:l2x": {
      "name": [
        {
          "name": "test4",
          "direction": "ingress",
          "incoming-interface": "ifp-0/1/12",
          "incoming-outer-vlan": 200,
          "outgoing-interface": "ifp-0/1/13",
          "match-type": "match-outer"
        }
      ]
    }
  }
```

Example 3: Local L2X Ingress VLAN Operations

```
{
    "rtbrick-config:l2x": {
      "name": [
        {
          "name": "test4",
          "direction": "ingress",
          "incoming-interface": "ifp-0/1/12",
          "incoming-outer-vlan": 200,
          "outgoing-interface": "ifp-0/1/13",
          "match-type": "match-outer",
          "ingress-vlan-operation": "single-vlan-add",
          "ingress-outer-vlan": 400
        }
      ]
    }
  }
```

Local L2X Bidirectional Configuration

This configuration enables redirecting traffic incoming (ingress) on a particular interface to another interface and vice versa on the same hardware device.

Syntax:

set l2x name <l2x-name> **bi-directional** <attribute> <value>

Attribute	Description
match-type <match-type>	(Mandatory) Match types with which traffic can be matched.

Attribute	Description
incoming-interface <incoming-interface>	(Mandatory) Incoming interface is where the traffic originates from.
outgoing-interface <outgoing-interface>	(Mandatory) Outgoing interface where the traffic is going to.
description <description>	(Optional) L2X description
egress-vlan-encapsulation <encapsulation>	(Optional) Egress VLAN encapsulation
incoming-inner-vlan <vlan-id>	(Optional) Incoming inner VLAN
incoming-outer-vlan <vlan-id>	(Optional) Incoming outer VLAN
ingress-outer-vlan <vlan-id>	(Optional) Outer VLAN at ingress side
ingress-vlan-encapsulation <encapsulation>	(Optional) Ingress VLAN encapsulation
outgoing-outer-vlan <vlan-id>	(Optional) Outgoing outer VLAN

Example 1: Local L2X Bidirectional Configuration

```
{
    "rtbrick-config:l2x": {
      "name": [
        {
          "name": "test2",
          "direction": "bi-directional",
          "incoming-interface": "ifp-0/1/64",
          "outgoing-interface": "ifp-0/1/66",
          "match-type": "match-untagged"
        }
      ]
    }
  }
```

Remote L2X Configuration

The following sections describe the remote L2X configurations.

Remote L2X Ingress Configuration

This configuration enables the remote L2X ingress side.

Syntax:

set l2x name <l2x-name> **ingress** <attribute> <value>

Attribute	Description
match-type <match-type>	(Mandatory) Match types with which traffic can be matched.
incoming-interface <incoming-interface>	(Mandatory) Incoming interface is where the traffic originates from.
ingress-vlan-operation <ingress-vlan-action>	(Optional) VLAN operation on ingress side outer VLAN
ingress-outer-vlan <vlan-id>	(Optional) Outer VLAN at ingress side
ingress-vlan-encapsulation <encapsulation>	(Optional) Ingress VLAN encapsulation value
nexthop4/nexthop6 <nexthop>	(Mandatory) Next-Hop address
lookup-instance <lookup-instance>	(Optional) Instance name
lookup-afi <lookup-afi>	(Optional) AFI value: ipv4 or ipv6
lookup-safi <lookup-safi>	(Optional) SAFI value: safi values are unicast, labeled-unicast
service-label <service_label>	(Mandatory) Service label value. NOTE: Supported MPLS label values are 0 - 1048575. The reserved MPLS label range is 0 - 15. In RBFS, BGP uses the label range 20000 - 100000. It is recommended to assign label values outside of these reserved ranges to avoid conflicts.

Example 1: Remote L2X Ingress Configuration with Port Match

```
{
    "rtbrick-config:l2x": {
      "name": [
        {
          "name": "test8",
          "direction": "ingress",
          "incoming-interface": "ifp-0/1/64",
          "nexthop4": "198.51.100.44",
          "lookup-instance": "default",
          "lookup-afi": "ipv4",
          "lookup-safi": "labeled-unicast",
          "service-label": 10000,
          "match-type": "match-any"
        }
      ]
    }
  }
```

Example 2: Remote L2X Ingress Configuration with VLAN match

```
{
    "rtbrick-config:l2x": {
      "name": [
        {
          "name": "test4",
          "direction": "ingress",
          "incoming-interface": "ifp-0/1/12",
          "incoming-outer-vlan": 100,
          "incoming-inner-vlan": 200,
          "nexthop4": "198.51.100.44",
          "lookup-instance": "default",
          "lookup-afi": "ipv4",
          "lookup-safi": "labeled-unicast",
          "service-label": 8000,
          "match-type": "match-outer-inner"
        }
      ]
    }
  }
```

Example 3: Remote L2X Ingress VLAN operations

```
{
    "rtbrick-config:l2x": {
      "name": [
        {
          "name": "test4",
          "direction": "ingress",
          "incoming-interface": "ifp-0/1/12",
          "incoming-outer-vlan": 100,
          "incoming-inner-vlan": 200,
          "nexthop4": "198.51.100.44",
          "lookup-instance": "default",
          "lookup-afi": "ipv4",
          "lookup-safi": "labeled-unicast",
          "service-label": 10000,
          "match-type": "match-outer-inner",
          "ingress-vlan-operation": "Single-Vlan-Delete"
        }
```

Remote L2X Egress Configuration

This configuration enables the remote L2X egress side.

Syntax:

set l2x name <l2x-name> **egress** <attribute> <value>

Attribute	Description
service-label <service_label>	(Mandatory) Service label value. NOTE: Supported MPLS label values are 0 - 1048575. The reserved MPLS label range is 0 - 15. In RBFS, BGP uses the label range between 20000 - 100000. It is recommended to assign a label value outside of these reserved ranges to avoid conflicts.
outgoing-interface <outgoing-interface>	(Mandatory) Interface where traffic is going to.
egress-vlan-operation <vlan-action>	(Optional) Outgoing VLAN operation
outgoing-outer-vlan <vlan-id>	(Optional) Outgoing outer VLAN

Example 1: Local L2X Egress Configuration

```
{
    "rtbrick-config:l2x": {
      "name": [
        {
          "name": "test4",
          "direction": "egress",
          "service-label": 10000,
          "outgoing-interface": "ifp-0/1/66"
        }
      ]
    }
  }
```

Example 2: Local L2X Egress Configuration with VLAN Operation

```
{
    "rtbrick-config:l2x": {
      "name": [
        {
          "name": "test4",
          "direction": "egress",
          "service-label": 10000,
          "outgoing-interface": "ifp-0/1/12",
          "egress-vlan-operation": "single-vlan-add",
          "outgoing-outer-vlan": 400
        }
      ]
    }
  }
```

3.1.3. L2X Operational Commands

The L2X show commands provide detailed information about the L2X operations.

L2X Show Commands

The L2X show commands display data from FIB local table. Therefore, local L2X with down ports or remote l2x with unresolved nexthop address are not displayed.

L2X Summary

The summary commands display L2X information in a tabular format. Key information is displayed in the summary output.

> The L2X name is truncated after certain length as space is less to display summary output. In such cases, you can use detail command output where full name is displayed.

Syntax:

show l2x <options>

Option	Description
-	Without any option, the commands displays all L2X information such as ingress L2X and egress L2X.
l2x-name <l2x-name>	Displays information for a specific L2X.
detail	Displays detailed L2X information for a specific L2X.
direction <direction>	Displays L2X information for a specified direction, where direction can be ingress, egress, or bi-directional.

Option	Description
local-interface <interface-name>	Displays L2X information for a specific LAG interface.
nexthop4 <nexthop>	Displays L2X information for the remote IPv4 address.
nexthop6 <nexthop>	Displays L2X information for the remote IPv6 address.
service-label <service_label>	Displays the L2X information for a specific service label.
type <type>	Displays detailed L2X information for a specific type and L2X.
statistics	Displays statistics for a specific L2X.

Example 1: Summary view of L2X information

```
supervisor@rtbrick: op> show l2x
Name                           Direction  Incoming Intf        Outgoing Intf/Next Hop    Outer VLAN
Inner VLAN      Service label
l2bsa-0/1/27/281479271677953   ingress    ifp-0/1/27           2001:db8:0:75::           64
Any          110011
l2bsa-0/1/27/281479271677953   egress     -                    ifp-0/1/27                -                -
120011
l2bsa-0/1/27/281479271677954   ingress    ifp-0/1/27           2001:db8:0:75::          65                Any
110012
l2bsa-0/1/27/281479271677954   egress     -                    ifp-0/1/27                -                -
120012
l2bsa-0/1/27/281479271677955   ingress    ifp-0/1/27           2001:db8:0:75::          66                Any
110013
l2bsa-0/1/27/281479271677955   egress     -                    ifp-0/1/27                -                -
120013
l2bsa-0/1/27/281479271677956   ingress    ifp-0/1/27           2001:db8:0:75::          67                Any
110014
l2bsa-0/1/27/281479271677956   egress     -                    ifp-0/1/27                -                -
120014
```

Example 2: Summary view of a specific L2X

```
supervisor@rtbrick: op> show l2x l2bsa-0/1/27/281479271677953
Name                           Direction  Incoming Intf        Outgoing Intf/Next Hop    Outer VLAN
Inner VLAN      Service label
l2bsa-0/1/27/281479271677953   ingress    ifp-0/1/27           2001:db8:0:75::           64               Any
110011
l2bsa-0/1/27/281479271677953   egress     -                    ifp-0/1/27                -                -
120011
```

Example 3: Summary view of a remote L2X

```
supervisor@rtbrick: op> show l2x type remote
Name                           Direction  Incoming Intf        Outgoing Intf/Next Hop    Outer VLAN
Inner VLAN      Service label
l2bsa_lag-1_11                 ingress    lag-1                2001:db8:0:85::           11               Any
120011
l2bsa_lag-1_11                 egress     -                    lag-1                     -                -
110011
```

```
l2bsa_lag-1_12                 ingress    lag-1                2001:db8:0:85::          12               Any
120012
l2bsa_lag-1_12                 egress     -                    lag-1                    -                -
110012
l2bsa_lag-1_13                 ingress    lag-1                2001:db8:0:85::          13               Any
120013
l2bsa_lag-1_13                 egress     -                    lag-1                    -                -
110013
l2bsa_lag-1_14                 ingress    lag-1                2001:db8:0:85::          14               Any
120014
l2bsa_lag-1_14                 egress     -                    lag-1                    -                -
110014
l2bsa_lag-1_15                 ingress    lag-1                2001:db8:0:85::          15               Any
120015
l2bsa_lag-1_15                 egress     -                    lag-1                    -                -
110015
l2bsa_lag-1_16                 ingress    lag-1                2001:db8:0:85::          16               Any
120016
```

Example 4: Summary view of L2X for a specific service label

```
supervisor@rtbrick: op> show l2x service-label 120011
Name                           Direction  Incoming Intf        Outgoing Intf/Next Hop   Outer VLAN    Inner
VLAN      Service label
l2bsa_lag-1_11                 ingress    lag-1                2001:db8:0:85::          11            Any
120011
supervisor@rtbrick: op>
```

Example 5: Summary view of the L2X information for a specified direction, where direction can be ingress, egress, or bi-directional.

```
supervisor@rtbrick: op> show l2x direction ingress
Name                           Direction  Incoming Intf        Outgoing Intf/Next Hop   Outer VLAN
Inner VLAN      Service label
l2bsa_lag-1_11                 ingress    lag-1                2001:db8:0:85::          11               Any
120011
l2bsa_lag-1_12                 ingress    lag-1                2001:db8:0:85::          12               Any
120012
l2bsa_lag-1_13                 ingress    lag-1                2001:db8:0:85::          13               Any
120013
l2bsa_lag-1_14                 ingress    lag-1                2001:db8:0:85::          14               Any
120014
l2bsa_lag-1_15                 ingress    lag-1                2001:db8:0:85::          15               Any
120015
l2bsa_lag-1_16                 ingress    lag-1                2001:db8:0:85::          16               Any
120016
l2bsa_lag-1_17                 ingress    lag-1                2001:db8:0:85::          17               Any
120017
l2bsa_lag-1_18                 ingress    lag-1                2001:db8:0:85::          18               Any
120018
l2bsa_lag-1_19                 ingress    lag-1                2001:db8:0:85::          19               Any
120019
l2bsa_lag-1_20                 ingress    lag-1                2001:db8:0:85::          20               Any
120020
l2bsa_lag-1_21                 ingress    lag-1                2001:db8:0:85::          21               Any
120021
l2bsa_lag-1_22                 ingress    lag-1                2001:db8:0:85::          22               Any
120022
l2bsa_lag-1_23                 ingress    lag-1                2001:db8:0:85::          23               Any
120023
l2bsa_lag-1_24                 ingress    lag-1                2001:db8:0:85::          24               Any
120024
l2bsa_lag-1_25                 ingress    lag-1                2001:db8:0:85::          25               Any
120025
l2bsa_lag-1_26                 ingress    lag-1                2001:db8:0:85::          26               Any
120026
l2bsa_lag-1_27                 ingress    lag-1                2001:db8:0:85::          27               Any
```

```
120027
```

Example 6: Summary view of L2X information for a specific LAG interface

```
supervisor@rtbrick: op> show l2x local-interface lag-1
Name                           Direction  Incoming Intf        Outgoing Intf/Next Hop    Outer VLAN
Inner VLAN      Service label
l2bsa_lag-1_11                 ingress    lag-1                2001:db8:0:85::            11
Any             120011
l2bsa_lag-1_12                 ingress    lag-1                2001:db8:0:85::            12
Any             120012
l2bsa_lag-1_13                 ingress    lag-1                2001:db8:0:85::            13
Any             120013
l2bsa_lag-1_14                 ingress    lag-1                2001:db8:0:85::            14
Any             120014
l2bsa_lag-1_15                 ingress    lag-1                2001:db8:0:85::            15
Any             120015
l2bsa_lag-1_16                 ingress    lag-1                2001:db8:0:85::            16
Any             120016
l2bsa_lag-1_17                 ingress    lag-1                2001:db8:0:85::            17
Any             120017
l2bsa_lag-1_18                 ingress    lag-1                2001:db8:0:85::            18
Any             120018
l2bsa_lag-1_19                 ingress    lag-1                2001:db8:0:85::            19
Any             120019
l2bsa_lag-1_20                 ingress    lag-1                2001:db8:0:85::            20
Any             120020
```

Example 7: Summary view of L2X information for a remote egress router

```
supervisor@rtbrick: op> show l2x nexthop4 198.51.100.103
Name                           Direction  Incoming Intf        Outgoing Intf/Next Hop    Outer VLAN
Inner VLAN      Service label
l2bsa_lag-1_11                 ingress    lag-1                198.51.100.103            11               Any
120011
l2bsa_lag-1_12                 ingress    lag-1                198.51.100.103            12               Any
120012
l2bsa_lag-1_13                 ingress    lag-1                198.51.100.103            13               Any
120013
l2bsa_lag-1_14                 ingress    lag-1                198.51.100.103            14               Any
120014
l2bsa_lag-1_15                 ingress    lag-1                198.51.100.103            15               Any
120015
l2bsa_lag-1_16                 ingress    lag-1                198.51.100.103            16               Any
120016
```

Example 8: Summary view of L2X information for a remote egress router

```
supervisor@rtbrick: op> show l2x  nexthop6 2001:db8:0:85::
Name                           Direction  Incoming Intf        Outgoing Intf/Next Hop    Outer VLAN
Inner VLAN      Service label
l2bsa_lag-1_11                 ingress    lag-1                2001:db8:0:85::           11               Any
120011
l2bsa_lag-1_12                 ingress    lag-1                2001:db8:0:85::           12               Any
120012
l2bsa_lag-1_13                 ingress    lag-1                2001:db8:0:85::           13               Any
120013
l2bsa_lag-1_14                 ingress    lag-1                2001:db8:0:85::           14               Any
120014
l2bsa_lag-1_15                 ingress    lag-1                2001:db8:0:85::           15               Any
120015
l2bsa_lag-1_16                 ingress    lag-1                2001:db8:0:85::           16               Any
120016
l2bsa_lag-1_17                 ingress    lag-1                2001:db8:0:85::           17               Any
120017
```

```
l2bsa_lag-1_18               ingress    lag-1              2001:db8:0:85::          18              Any
120018
l2bsa_lag-1_19               ingress    lag-1              2001:db8:0:85::          19              Any
120019
l2bsa_lag-1_20               ingress    lag-1              2001:db8:0:85::          20              Any
120020
l2bsa_lag-1_21               ingress    lag-1              2001:db8:0:85::          21              Any
120021
```

Example 9: L2X information in detailed format

```
supervisor@rtbrick: op> show l2x detail
L2X name: l2bsa_lag-1_11
  Direction: ingress
  Status: Download success
  Incoming interface: lag-1
  Service label: 120011
  Subtype: Incoming Port - Outer Vlan - Any Inner Vlan Match
  Incoming outer VLAN: 11
  Incoming inner VLAN: Any
  Ingress vlan operation:
    Vlan operation: Swap-Outer-Vlan
    Outer vlan: 64
  NextHop:
    NextHop IP: 2001:db8:0:85::
    Lookup instance: default
    Lookup AFI: ipv6
    Lookup SAFI: labeled-unicast
    NextHop type: Remote ingress cross connect
    NextHop action: mpls label push
  Egress vlan operation:
```

Example 10: Detailed L2X information for a specific L2X

```
supervisor@rtbrick: op> show l2x test1 detail
L2X name: test1
  Direction: ingress
  Status: Download success
  Incoming interface: ifp-0/0/4
  Outgoing interface: ifp-0/0/10
  Subtype: Incoming Port - Any Vlan Match
  Incoming outer VLAN: Any
  Incoming inner VLAN: Any
  Ingress vlan operation:
  NextHop:
    NextHop type: Local egress cross connect
    NextHop action: No vlan manipulation - l2 forward
  Egress vlan operation:
```

Example 11: Detailed L2X information for a specific type and L2X

```
supervisor@rtbrick: op> show l2x type local test1 detail
L2X name: test1
  Direction: ingress
  Status: Download success
  Incoming interface: ifp-0/0/4
  Outgoing interface: ifp-0/0/10
  Subtype: Incoming Port - Any Vlan Match
  Incoming outer VLAN: Any
  Incoming inner VLAN: Any
  Ingress vlan operation:
  NextHop:
    NextHop type: Local egress cross connect
    NextHop action: No vlan manipulation - l2 forward
  Egress vlan operation:
```

Example 12: Detailed L2X information for a specific direction and L2X

```
supervisor@rtbrick: op> show l2x direction egress test2 detail
L2X name: test2
  Direction: egress
  Status: Download success
  Outgoing interface: ifp-0/0/4
  Service label: 1234
  Subtype: Service Label Match
  Incoming outer VLAN: -
  Incoming inner VLAN: -
  Ingress vlan operation:
  NextHop:
    NextHop type: Remote egress cross connect
    NextHop action: No vlan manipulation - l2 forward
  Egress vlan operation:
```

Example 13: Statistics for all installed L2X

```
supervisor@rtbrick: op> show l2x statistics

L2X Name: l2x-test1/0
    Physical Interface Name: ifp-0/0/4
    Logical Interface Type: L2x ingress vlan interface
    Port-Mapping-Core: 0
    Vlan-Port-ID: 1149251592
    MPLS-Port-ID: N/A
    Counters:
        In-Forward-Packets: 57
        In-Forward-Bytes: 5700
        In-Drop-Packets: 0
        In-Drop-Bytes: 0
        Out-Forward-Packets: 0
        Out-Forward-Bytes: 0
        Out-Drop-Packets: 0
        Out-Drop-Bytes: 0
L2X Name: l2x-d3b529d74770f91fb2acf5e38da70eb9213473dd7e996c6a
    Physical Interface Name: ifp-0/0/10
    Logical Interface Type: L2x egress vlan interface
    Port-Mapping-Core: 0
    Vlan-Port-ID: 1149251591
    MPLS-Port-ID: N/A
    Counters:
        In-Forward-Packets: 0
        In-Forward-Bytes: 0
```

```
          In-Drop-Packets: 0
          In-Drop-Bytes: 0
          Out-Forward-Packets: 0
          Out-Forward-Bytes: 0
          Out-Drop-Packets: 0
          Out-Drop-Bytes: 0
```

L2X Clear Commands

Clear commands allow to reset operational states.

L2X Statistics

This commands resets L2X statistics.

Syntax:

clear l2x statistics <l2x-name>

Attribute	Description
-	Without any option, the command clears all L2X statistics.
<l2x-name>	L2X name.

Example:

```
supervisor@rtbrick: op> clear l2x statistics l2x-test1/0
```

3.2. EVPN-VPWS

3.2.1. EVPN-VPWS Overview

Ethernet Virtual Private Network (EVPN) is a Layer 2 internetworking technology similar to BGP/MPLS IP VPN. EVPN uses extended BGP reachability information and advertisements between different Layer 2 networks at various sites in the control plane.

The EVPN Virtual Private Wire Service (VPWS) is a point-to-point (P2P) service that is built on the EVPN service architecture. EVPN-VPWS uses MPLS tunnels to traverse the backbone network. It offers a Layer 2 packet forwarding mode that connects access circuits (ACs) as per the specifications of RFC 8214.

Supported Standards

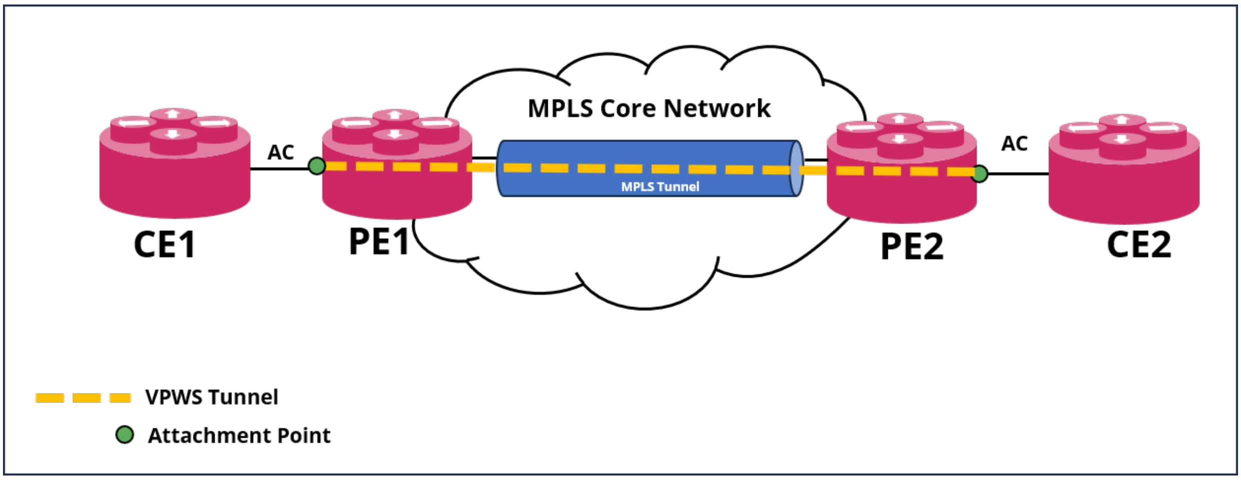
RFC Number	Description
7432	BGP MPLS-Based Ethernet VPN
8214	Virtual Private Wire Service Support in Ethernet VPN

EVPN VPWS Network Model

As shown in the figure below, an EVPN VPWS network contains the following building blocks:

- Customer edge (CE)—Customer device directly connected to the service provider network.
- Provider edge (PE)—Service provider device connected to CEs. PEs provide access to the EVPN VPWS network and forward traffic between customer network sites by using public tunnels.
- Attachment circuit (AC)—A physical or virtual link between a CE and a PE.
- Pseudowire (PW)—A virtual bidirectional connection between two PEs. A PW comprises a pair of virtual links in opposite directions.
- MPLS transport tunnel—A connection that carries one or more PWs across the MPLS core or IP backbone, such as an MPLS tunnel.
- Cross-connect—A connection formed by two physical or virtual circuits, such as ACs and PWs, that switches packets between them.

The figure below shows the protocol packet exchange process in the EVPN VPWS.



PE1 and PE2 are each configured with an EVPN VPWS instance. After the PE receives packets from the AC, it adds the PW label and sends them to the peer PE through the MPLS transport tunnel. After the other PE (PE2) receives the packet via the MPLS transport path, it removes the PW label of the packets and forwards the packets to the AC bound to the PW.

Supported Platforms

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

Unsupported Features

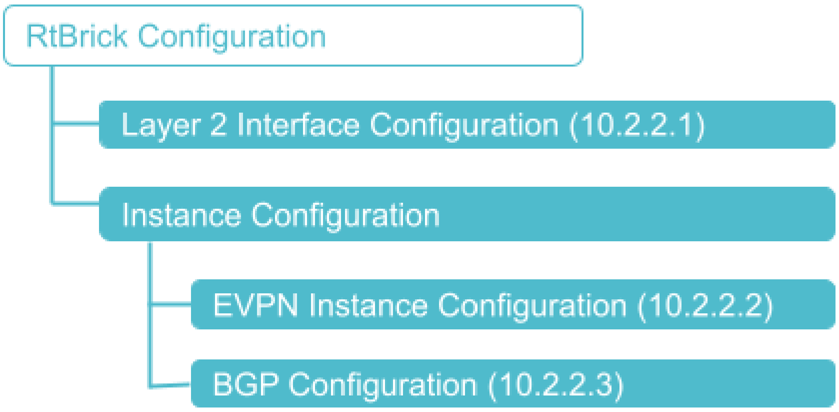
The following features are not supported for EVPN VPWS:

- Policy filtering options (at Address Family/Peer-Group/Instance levels)
- LAG interface as an attachment circuit
- LAG interface as an LSP MPLS path
- Route Reflection
- Add-Path
- VLAN change configuration on L2 IFL

3.2.2. EVPN VPWS Configuration

Configuration Hierarchy

The diagram illustrates the EVPN-VPWS configuration hierarchy. All EVPN VPWS configurations are done within an instance, such as the default instance or an EVPN service instance.


RtBrick Configuration
Layer 2 Interface Configuration (10.2.2.1)
Instance Configuration
EVPN Instance Configuration (10.2.2.2)
BGP Configuration (10.2.2.3)

Configuration Syntax and Commands

The following sections describe the EVPN-VPWS configuration syntax and commands.

Layer 2 Interface Configuration

EVPN-VPWS supports the configuration of Layer 2 logical interfaces.

Syntax:

set interface <name> <attribute> <value>

Attribute	Description
<name>	Specify the name of the interface. Example: ifp-0/0/1.
unit <unit-id>	Create a logical interface (also referred to as a sub-interface) under the physical interface
unit <unit-id> interface-type l2vpn-vpws	Specify the type of the L2VPN interface.
unit <unit-id> vlan <outer-vlan-id>	Outer VLAN ID.
unit <unit-id> instance <instance>	Assign the logical interface to an instance.

The following example shows an untagged interface configuration:

```
set interface ifp-0/0/17 unit 0
set interface ifp-0/0/17 unit 0 interface-type l2vpn-vpws
set interface ifp-0/0/17 unit 0 instance evpn-vpws-vrf1
```

```
commit
```

The following example shows a single-tagged interface configuration:

```
set interface ifp-0/0/17 unit 100
set interface ifp-0/0/17 unit 100 interface-type l2vpn-vpws
set interface ifp-0/0/17 unit 100 instance evpn-vpws-vrf2
set interface ifp-0/0/17 unit 100 vlan 100
commit
```

The following example shows a double-tagged interface configuration:

```
set interface ifp-0/0/17 unit 200
set interface ifp-0/0/17 unit 200 interface-type l2vpn-vpws
set interface ifp-0/0/17 unit 200 instance evpn-vpws-vrf3
set interface ifp-0/0/17 unit 200 vlan 200
set interface ifp-0/0/17 unit 200 inner-vlan 201
commit
```

EVPN Instance Configuration

An EVPN Instance (EVI) is a routing and forwarding instance of EVPN that covers all the participating PE routers in a VPN. EVI is configured per customer on the PE routers. Each EVI has a unique route distinguisher and one or more route targets.

Syntax:

set instance <name> <attribute> <value>

Attribute	Description
<name>	A unique name for the EVPN routing instance.
address-family <afi>	Address family identifier (AFI). Supported value: l2vpn
address-family <afi> <safi>	Subsequent address family identifier (SAFI). Supported values: evpn-vpws ℹ evpn-vpws needs to be enabled on the VRF instance.
protocol	Specifies the routing protocol

Attribute	Description
route-distinguisher <as-number | ipv4-address:id>	The route distinguisher (RD) uniquely defines routes within an IPv4 network. PE routers use route distinguishers to identify which VPN a packet belongs to. Supported formats are <as-number:id> or <ipv4-address:id>.
ipv4-router-id <ipv4-router-id>	The router ID of the routing instance.
route-target (import | export) <rt-value>	Route targets (RT) are used to transfer routes between VPN instances. The RT identifies a subset of routes that should be imported to or exported from a particular VPN instance. You can configure an RT for importing or exporting routes or both.

In the following configuration, VRF instance AFI has been set to AFI: l2vpn and SAFI: evpn-vpws.

```
set instance evpn-vpws-vrf1
set instance evpn-vpws-vrf1 ipv4-router-id 192.1.6.3
set instance evpn-vpws-vrf1 route-distinguisher 192.1.6.3:65006
set instance evpn-vpws-vrf1 address-family l2vpn evpn-vpws
set instance evpn-vpws-vrf1 address-family l2vpn evpn-vpws route-target import
target:192.1.6.0:65006
set instance evpn-vpws-vrf1 address-family l2vpn evpn-vpws route-target export
target:192.1.6.0:65006
commit
```

BGP Configuration

BGP L2VPN VFT (Virtual Forwarding Table) Configuration

Syntax:

set instance <name> **protocol bgp** <attribute> <value>

Attribute	Description
<name>	Name of the routing instance
address-family <afi>	Address family identifier (AFI). Supported value: l2vpn

Attribute	Description
address-family <afi> <safi>	Subsequent address family identifier (SAFI). Supported value: evpn-vpws ℹ evpn-vpws needs to be enabled on the VRF instance.
local-as <as-number>	The AS number in four-byte format. The numbers allowed are from 1 to 4294967285.
interface <name>	Interface that is bound to L2VPN
interface <name> local-service-id <local-service-id>	Specify the local service ID that is used to establish an EVPN PW between two PEs
interface <name> remote-service-id <remote-service-id>	Specify the remote service ID that is used to establish an EVPN PW between two PEs

The following example configures evpn-vpws-vrf1 instance for BGP L2VPN.

```
set instance evpn-vpws-vrf1 protocol bgp local-as 65006
set instance evpn-vpws-vrf1 protocol bgp address-family l2vpn evpn-vpws
set instance evpn-vpws-vrf1 protocol bgp address-family l2vpn evpn-vpws interface
ifl-0/0/17/0
set instance evpn-vpws-vrf1 protocol bgp address-family l2vpn evpn-vpws interface
ifl-0/0/17/0 local-service-id 100
set instance evpn-vpws-vrf1 protocol bgp address-family l2vpn evpn-vpws interface
ifl-0/0/17/0 remote-service-id 101
commit
```

BGP L2VPN EVPN Configuration

Configuring the BGP L2VPN EVPN Address Family

Syntax:

set instance <name> **protocol bgp address-family l2vpn evpn**

Attribute	Description
<name>	Name of the routing instance

To configure BGP L2VPN EVPN on the default instance, enter the following command:

```
set instance default protocol bgp address-family l2vpn evpn
commit
```

Configuring Address Families for Peer Groups

Syntax:

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

Attribute	Description
<afi>	Address family identifier (AFI). Supported value: l2vpn.
<safi>	Subsequent address family identifier (SAFI). Supported value: evpn
extended-nexthop	Enable extended-next-hop encoding for BGP peer groups to allow the transfer of IPv4 prefixes over an IPv6 connection.
update-nexthop (ipv4-address | ipv6-address) <address>	BGP nexthop address for routes advertised to this peer group

To configure BGP L2VPN Peer Group on the default instance, enter the following command:

```
set instance default protocol bgp peer-group spine address-family l2vpn evpn
commit
```

3.2.3. EVPN VPWS Operational Commands

EVPN VPWS Show Commands

The following show commands display the EVPN VPWS-related information.

BGP Summary

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

Syntax:

show bgp summary <option>

Attribute	Description
Option	Description
-	Without any option, the command displays the information for all instances.

Example: BGP summary for instance evpn-vpws-vrf1

```
supervisor@rtbrick: op> show bgp summary instance evpn-vpws-vrf1
Instance: evpn-vpws-vrf1
  General information
    Hostname: , Domain name:
    Local AS: 65006, Version: 4
    Local preference: 100, eBGP Protocol preference: 20, iBGP Protocol preference:
200
    Router ID: 192.168.6.10, Cluster ID: 192.168.6.10
  Capabilities
    Route refresh: True, AS4: True, Graceful restart: False, L2VPN EVPN-VPWS:True
  Best route selection
    Always compare MED: False, Ignore as path: False
    Ignore local preference: False, Ignore origin: False
    Ignore MED: False, Ignore route source: False
    Ignore router ID: False, Ignore uptime: True
    Ignore cluster length: False, Ignore peer IP: False
    Route select parameter: 0
  Timers
    Connect retry: 30s, Keepalive: 30s, Holdtime: 90s
  Statistics
    Peers configured: 0, Peers auto discovery: 0
      Peers in idle          : 0
      Peers in connect       : 0
      Peers in active        : 0
      Peers in opensent      : 0
      Peers in openconfirm   : 0
      Peers in established   : 0
```

BGP Peer

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

Syntax:

show bgp peer <option>

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

Example 1: BGP Peer Information for instance default

```
supervisor@rtbrick: op> show bgp peer instance default
Instance: default
  Peer     Remote AS    State         Up/Down Time    PfxRcvd         PfxSent
  spine1   4200000000   Established   0d:00h:42m:53s  5               14
supervisor@rtbrick: op>
```

Example 2: Detailed view of BGP Peer Information

```
supervisor@rtbrick: op> show bgp peer detail
Peer: spine1, Peer IP: fe80::9a19:2cff:fe99:4701, Remote AS: 4200000000, Local:
fe80::9a19:2cff:fe36:3e03, Local AS: 4200000004, Any AS: False
  Type: ebgp, State: Established, Up/Down Time: 0d:01h:00m:40s
  Discovered on interface: ifl-0/0/2/1
  Last transition: Thu Apr 18 04:50:56 GMT +0000 2024, Flap count: 0
  Peer ID        : 192.1.0.1, Local ID  : 192.1.0.4
  Instance       : default, Peer group: spine
  6PE enabled    : False
  Timer values:
    Peer keepalive : 30s, Local keepalive: 30s
    Peer holddown  : 90s, Local holddown : 90s
    Connect retry  : 30s
  Timers:
    Connect retry timer : 0s
    keepalive timer     : expires in 21s 796304us
    Holddown timer      : expires in 1m 13s 560627us
  NLRIs:
    Sent           : ['l2vpn-evpn', 'ipv4-unicast', 'ipv6-unicast', 'ipv4-vpn-
unicast', 'ipv6-vpn-unicast', 'ipv4-vpn-multicast', 'l2vpn-vpls', 'ipv4-labeled-
unicast', 'ipv6-labeled-unicast']
    Received       : ['l2vpn-evpn', 'ipv4-unicast', 'ipv6-unicast', 'ipv4-vpn-
unicast', 'ipv6-vpn-unicast', 'ipv4-vpn-multicast', 'l2vpn-vpls', 'ipv4-labeled-
unicast', 'ipv6-labeled-unicast']
    Negotiated     : ['l2vpn-evpn', 'ipv4-unicast', 'ipv6-unicast', 'ipv4-vpn-
unicast', 'ipv6-vpn-unicast', 'ipv4-vpn-multicast', 'l2vpn-vpls', 'ipv4-labeled-
unicast', 'ipv6-labeled-unicast']
  Capabilities:
    Addpath sent                 : None
    Addpath received             : None
    Addpath negotiated           : None
    Extended nexthop sent        : ['ipv4-unicast', 'ipv4-vpn-unicast', 'ipv4-
vpn-multicast', 'ipv4-labeled-unicast']
    Extended nexthop received    : ['ipv4-unicast', 'ipv4-labeled-unicast',
'ipv4-vpn-multicast', 'ipv4-vpn-unicast']
    Extended nexthop negotiated  : ['ipv4-unicast', 'ipv4-labeled-unicast',
'ipv4-vpn-multicast', 'ipv4-vpn-unicast']
    Capabilities:
      Feature                     Sent            Received        Negotiated
      Route refresh               True            True            True
      4 byte AS                   True            True            True
      Graceful restart            False           False           False
      Link local only             False           False           False
  Prefix Limit:
  End of RIB:
    Address family              Sent                                Received
    IPv4 unicast                Thu Apr 18 04:50:59 GMT +0000 2024  Thu Apr 18
04:50:59 GMT +0000 2024
    IPv4 labeled-unicast        Thu Apr 18 04:50:59 GMT +0000 2024  Thu Apr 18
04:50:59 GMT +0000 2024
    IPv6 unicast                Thu Apr 18 04:50:59 GMT +0000 2024  Thu Apr 18
04:50:59 GMT +0000 2024
    IPv6 labeled-unicast        Thu Apr 18 04:50:59 GMT +0000 2024  Thu Apr 18
04:50:59 GMT +0000 2024
    IPv4 VPN-unicast            Thu Apr 18 04:50:59 GMT +0000 2024  Thu Apr 18
04:50:59 GMT +0000 2024
    IPv6 VPN-unicast            Thu Apr 18 04:50:59 GMT +0000 2024  Thu Apr 18
04:50:59 GMT +0000 2024
```

```
    IPv4 VPN-multicast             Thu Apr 18 04:50:59 GMT +0000 2024  Thu Apr 18
04:50:59 GMT +0000 2024
    L2VPN VPLS                     Thu Apr 18 04:50:59 GMT +0000 2024  Thu Apr 18
04:50:59 GMT +0000 2024
    L2VPN EVPN                     Thu Apr 18 04:50:59 GMT +0000 2024  Thu Apr 18
04:50:59 GMT +0000 2024
  Message stats:
    Session stats:
      Direction   Open         Update       Keepalive    Notify       Route
refresh
      Input       0            0            72           0            0
      Output      0            0            71           0            0
    Total stats:
      Input       0            0            72           0            0
      Output      0            0            71           0            0
    Route stats:
      Address family                 Received     Sent         Prefix limit Idle
timeout
      IPv4 unicast                   0            0            0            0
      IPv4 labeled-unicast           0            0            0            0
      IPv6 unicast                   0            0            0            0
      IPv6 labeled-unicast           0            0            0            0
      IPv4 VPN-unicast               0            0            0            0
      IPv6 VPN-unicast               0            0            0            0
      IPv4 VPN-multicast             0            0            0            0
      L2VPN VPLS                     0            0            0            0
      L2VPN EVPN                     0            0            0            0
<...>
```

Example 3: BGP Peer Statistics for a specific peer

```
supervisor@rtbrick: op> show bgp peer statistics peer address
fe80::781a:c6ff:fec0:1
Instance: default
  Peer                           AFI     SAFI               PfxRcvd    PfxSent
  spine1                         ipv4    unicast            0          0
                                 ipv4    labeled-unicast    0          0
                                 ipv6    unicast            1          4
                                 ipv6    labeled-unicast    1          4
                                 ipv4    vpn-unicast        0          0
                                 ipv6    vpn-unicast        0          0
                                 ipv4    multicast          0          0
                                 ipv4    vpn-multicast      0          0
                                 l2vpn   evpn               3          6
```

BGP RIB-in

This command displays the received routes.

Syntax:

show bgp rib-in <option>

Option	Description
-	Without any option, the command displays information on the received BGP routing table on all instances in a summary table format.
<afi>	BGP routing table summary for the given address family (AFI), all sub-address families and all instances. Supported AFI values are 'ipv4', 'ipv6' and l2vpn.
<afi> <safi>	BGP routing table summary for the given address family (AFI) and sub-address family (SAFI), and all instances. Supported SAFI values are evpn, 'labeled-unicast', 'unicast', 'vpn-multicast', 'vpn-unicast', and evpn-vpws.
<afi> <safi> detail	Detailed list view of the BGP routing table for the given address family (AFI) and sub-address family (SAFI), and all instances.
<afi> <safi> <prefix>	BGP routing table entry for the given prefix and all instances.
<afi> <safi> instance <instance-name>	BGP routing table summary for the given AFI, SAFI, and instance.
<afi> <safi> instance <instance-name> detail	Detailed list view of BGP routing table for the given AFI, SAFI, and instance.
<afi> <safi> instance <instance-name> <prefix>	BGP routing table entry for the given prefix and instance.
<afi> <safi> peer <name> / peer address <ip>	Peer name or address

Example 1: Summary view of the BGP rib-in where you can find received information for the EVPN VPWS instances like evpn-vpws-vrf1, evpn-vpws-vrf2, and evpn-vpws-vrf3.

```
supervisor@rtbrick: op> show bgp rib-in
Instance: default, AFI: ipv6, SAFI: unicast
 Hostname: Local, Peer IP: ::, Source IP: None, Received routes: 3
    Prefix                                        Next Hop                            MED        Local
Preference  AS Path    Status
    192:1::3/128                                                                      0          100
-          Valid
    192:1::5/128                                                                      0          100
-          Valid
    192:1::6/128                                                                      0          100
-          Valid
```

```
 Hostname: spine1, Peer IP: fe80::781a:c6ff:fec0:1, Source IP: fe80::7857:d6ff:fec0:0, Received routes: 1
    Prefix                                        Next Hop                            MED        Local
Preference  AS Path    Status
    192:1::1/128                                  fe80::781a:c6ff:fec0:1              0          -
4200000000 Valid
Instance: default, AFI: ipv6, SAFI: labeled-unicast
 Hostname: Local, Peer IP: ::, Source IP: None, Received routes: 3
    Prefix                                        Next Hop                            MED        Local
Preference  AS Path    Status
    192:1::3/128                                                                      0          100
-          Valid
    192:1::5/128                                                                      0          100
-          Valid
    192:1::6/128                                                                      0          100
-          Valid
 Hostname: spine1, Peer IP: fe80::781a:c6ff:fec0:1, Source IP: fe80::7857:d6ff:fec0:0, Received routes: 1
    Prefix                                        Next Hop                            MED        Local
Preference  AS Path    Status
    192:1::1/128                                  fe80::781a:c6ff:fec0:1              0          -
4200000000 Valid
Instance: evpn-vpws-vrf1, AFI: l2vpn, SAFI: evpn-vpws
 Hostname: Local, Peer IP: 0.0.0.0, Source IP: None, Received routes: 1
    Prefix                                        Next Hop                            MED        Local
Preference  AS Path    Status
    00.00.00.00.00.00.00.00.00.00:100/112                                             0          100
-          Valid
Instance: evpn-vpws-vrf2, AFI: l2vpn, SAFI: evpn-vpws
 Hostname: Local, Peer IP: 0.0.0.0, Source IP: None, Received routes: 1
    Prefix                                        Next Hop                            MED        Local
Preference  AS Path    Status
    00.00.00.00.00.00.00.00.00.00:110/112                                             0          100
-          Valid
Instance: evpn-vpws-vrf3, AFI: l2vpn, SAFI: evpn-vpws
 Hostname: Local, Peer IP: 0.0.0.0, Source IP: None, Received routes: 1
    Prefix                                        Next Hop                            MED        Local
Preference  AS Path    Status
    00.00.00.00.00.00.00.00.00.00:200/112                                             0          100
-          Valid
Instance: default, AFI: l2vpn, SAFI: evpn
 Hostname: spine1, Peer IP: fe80::781a:c6ff:fec0:1, Source IP: fe80::7857:d6ff:fec0:0, Received routes: 3
    Prefix                                        Next Hop                            MED        Local
Preference  AS Path    Status
    00.00.00.00.00.00.00.00.00.00:101/112         192:1::1                            0          -
4200000000 Valid
    00.00.00.00.00.00.00.00.00.00:111/112         192:1::1                            0          -
4200000000 Valid
    00.00.00.00.00.00.00.00.00.00:201/112         192:1::1                            0          -
4200000000 Valid
```

Example 2: Summary view of the BGP rib-in where you can find received information for address family l2vpn vpws.

```
supervisor@rtbrick: op> show bgp rib-in l2vpn evpn
Instance: default, AFI: l2vpn, SAFI: evpn
 Hostname: spine1, Peer IP: fe80::9a19:2cff:fe99:4701, Source IP: fe80::9a19:2cff:fe36:3e03, Received routes:
3
    Prefix                                        Next Hop                            MED        Local
Preference  AS Path    Status
    00.00.00.00.00.00.00.00.00.00:101/112         192:1::1                            0          -
4200000000, 4200000003 Valid
    00.00.00.00.00.00.00.00.00.00:111/112         192:1::1                            0          -
4200000000, 4200000003 Valid
    00.00.00.00.00.00.00.00.00.00:201/112         192:1::1                            0          -
4200000000, 4200000003 Valid
```

BGP FIB

The 'show bgp fib' commands display the BGP forwarding table.

Syntax:

show bgp fib <option>

Option	Description
-	Without any option, the commands display the BGP forwarding table for all address families and all instances in a summary table format.
<afi>	BGP forwarding table summary for the given address family (AFI), all sub-address families and all instances. Supported AFI values are l2vpn, 'ipv4' and 'ipv6'.
<afi> <safi>	BGP forwarding table summary for the given address family (AFI) and sub-address family (SAFI), and all instances. Supported SAFI values are 'unicast', 'labeled-unicast', 'vpn-multicast', 'vpn-unicast', and evpn-vpws.
<afi> <safi> detail	Detailed list view of the BGP forwarding table for the given address family (AFI) and sub-address family (SAFI), and all instances.
<afi> <safi> <prefix>	BGP forwarding table entry for the given prefix and all instances.
<afi> <safi> instance <instance-name>	BGP forwarding table summary for the given AFI, SAFI, and instance.
<afi> <safi> instance <instance-name> detail	Detailed list view of BGP forwarding table for the given AFI, SAFI, and instance.
<afi> <safi> instance <instance-name> <prefix>	BGP forwarding table entry for the given prefix and instance.

```
supervisor@rtbrick: op> show bgp fib
Instance: default, AFI: ipv6, SAFI: unicast
  Prefix                                        Preference      Out Label            Next Hop
  192:1::1/128                                  20              -                    fe80::781a:c6ff:fec0:1
Instance: default, AFI: ipv6, SAFI: labeled-unicast
  Prefix                                        Preference      Out Label            Next Hop
  192:1::1/128                                  20              -                    fe80::781a:c6ff:fec0:1
Instance: evpn-vpws-vrf1, AFI: l2vpn, SAFI: evpn-vpws
  Prefix                                        Preference      Out Label            Next Hop
  00.00.00.00.00.00.00.00.00.00:101/112         20              20004,bos:1          192:1::1
```

```
Instance: evpn-vpws-vrf2, AFI: l2vpn, SAFI: evpn-vpws
  Prefix                                        Preference      Out Label            Next Hop
  00.00.00.00.00.00.00.00.00.00:111/112         20              20005,bos:1          192:1::1
Instance: evpn-vpws-vrf3, AFI: l2vpn, SAFI: evpn-vpws
  Prefix                                        Preference      Out Label            Next Hop
  00.00.00.00.00.00.00.00.00.00:201/112         20              20006,bos:1          192:1::1
```

BGP RIB-out

This command displays the routes that were advertised to peers.

Syntax:

show bgp rib-out <option>

Option	Description
-	Without any option, the command displays advertised BGP routes for all instances.
<afi>	BGP routing table summary for the given address family (AFI), all sub-address families and all instances. Supported AFI values are l2vpn, 'ipv4' and 'ipv6'.
<afi> <safi>	BGP routing table summary for the given address family (AFI) and sub-address family (SAFI), and all instances. Supported SAFI values are evpn, 'unicast', 'labeled-unicast', 'multicast', and 'vpn-unicast'.
<afi> <safi> detail	Detailed list view of the BGP routing table for the given address family (AFI) and sub-address family (SAFI), and all instances.
<afi> <safi> <prefix>	BGP routing table entry for the given prefix and all instances.
<afi> <safi> instance <instance-name>	BGP routing table summary for the given AFI, SAFI, and instance.
<afi> <safi> instance <instance-name> detail	Detailed list view of BGP routing table for the given AFI, SAFI, and instance.
<afi> <safi> instance <instance-name> <prefix>	BGP routing table entry for the given prefix and instance.
<afi> <safi> peer <name> / peer address <ip>	Peer name or address

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

```
supervisor@rtbrick: op> show bgp rib-out
Instance: default, AFI: ipv6, SAFI: unicast
  Peer-group: spine, Sent routes: 4
    Prefix                                        MED             Local Preference  Origin          Next Hop
AS Path
    192:1::1/128                                  0               -                 Incomplete      -
4200000003, 4200000000
    192:1::3/128                                  0               -                 Incomplete      -
4200000003
    192:1::5/128                                  0               -                 Incomplete      -
4200000003
    192:1::6/128                                  0               -                 Incomplete      -
4200000003
Instance: default, AFI: ipv6, SAFI: labeled-unicast
  Peer-group: spine, Sent routes: 4
    Prefix                                        MED             Local Preference  Origin          Next Hop
AS Path
    192:1::1/128                                  0               -                 Incomplete      -
4200000003, 4200000000
    192:1::3/128                                  0               -                 Incomplete      -
4200000003
    192:1::5/128                                  0               -                 Incomplete      -
4200000003
    192:1::6/128                                  0               -                 Incomplete      -
4200000003
#Instance: default, AFI: l2vpn, SAFI: evpn
#  Peer-group: spine, Sent routes: 6
    Prefix                                        MED             Local Preference  Origin          Next Hop
AS Path
    00.00.00.00.00.00.00.00.00.00:100/112         0               -                 Incomplete      192:1::3
4200000003
    00.00.00.00.00.00.00.00.00.00:101/112         0               -                 Incomplete      192:1::3
4200000003, 4200000000
    00.00.00.00.00.00.00.00.00.00:110/112         0               -                 Incomplete      192:1::3
4200000003
    00.00.00.00.00.00.00.00.00.00:111/112         0               -                 Incomplete      192:1::3
4200000003, 4200000000
    00.00.00.00.00.00.00.00.00.00:200/112         0               -                 Incomplete      192:1::3
4200000003
    00.00.00.00.00.00.00.00.00.00:201/112         0               -                 Incomplete      192:1::3
4200000003, 4200000000
```

BGP L2VPN

This command displays the BGP L2VPN information.

Syntax:

show bgp l2vpn <option>

Option	Description
pseudowire	Displays pseudowire information for all instances.
pseudowire instance <name>	Displays pseudowire information for the specified instance.

Example: Display L2VPN pseudowire information for instance evpn-vpws-vrf1

```
supervisor@rtbrick: op> show bgp l2vpn pseudowire instance evpn-vpws-vrf1
   Instance: evpn-vpws-vrf1 AFI: l2vpn, SAFI: evpn-vpws
   Route Distinguisher: 192.168.6.10:65006
   Number of local interfaces: 1
    Interface name  Prefix                                     Status         Local SID   Remote SID   Pop
Label            Push Label
    ifl-0/0/3/0  00.00.00.00.00.00.00.00.00.00:100/112      up             100         101
label:20004,bos:1    label:20004,bos:1
```

4. Subscriber Management

4.1. Subscriber Management

4.1.1. Subscriber Management Overview

The modular, scalable subscriber management that RtBrick calls the next-generation access infrastructure (ng-access) provides support for protocols such as PPPoE, L2TPv2, DHCPv4/v6, and RADIUS.

The subscriber management infrastructure provides the next generation of internet access protocols designed for carrier-grade services in regard to scalability and robustness.

One of the challenges for carrier networks is interworking with numerous client devices and various vendors which require a well-implemented, industry-proven access protocol stack, including support for all relevant RFCs.

This implementation is designed to be a set of distributed services for increased scaling and stability.

Supported Platforms

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

Subscriber Management Daemons

There are four main daemons in the RtBrick distributed access architecture:

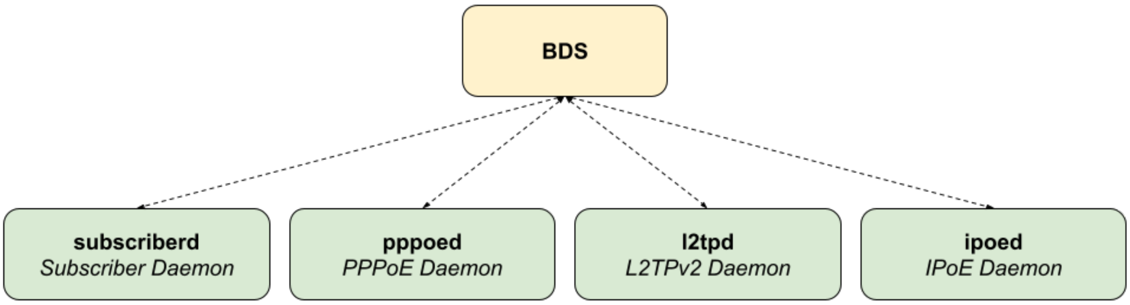


Figure 1. The Next Generation Access (ngaccess) Infrastructure

The subscriber daemon (subscriberd) is the central application, keeping the current subscriber state as well as being responsible for Authentication, Authorization, and Accounting (AAA).

- *subscriberd* is for subscriber management and AAA (which can be local, through RADIUS, or other methods)
- *pppoed* is to handle PPPoE and PPP sessions
- *l2tpd* is for L2TPv2 tunnel and session handling
- *ipoed* is for IPoE (IP-over-Ethernet aka DHCP) subscriber handling

This document describes the RBFS subscriber management implementation and configuration. The term subscriber describes an access user or session from a higher level decoupled from underlying protocols like PPPoE or IPoE.

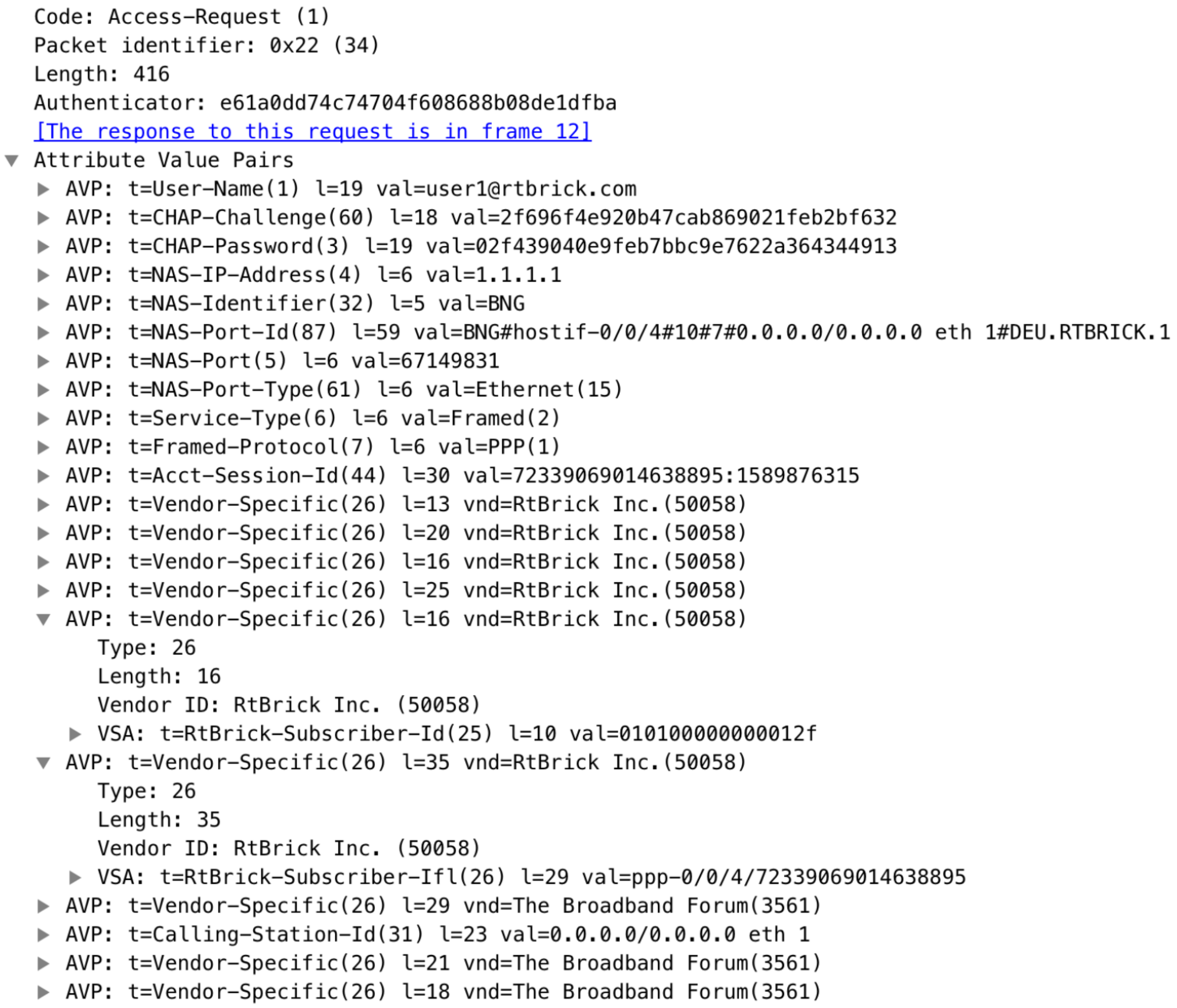
Subscribers in RBFS can be managed locally or remotely via RADIUS. Each subscriber is uniquely identified by a 64-bit number called subscriber-id.

Remote Authentication Dial-In User Service (RADIUS)

Remote Authentication Dial-In User Service (RADIUS) is a networking protocol that provides centralized Authentication, Authorization, and Accounting (AAA) management for all types of subscribers (PPPoE, or IPoE). RADIUS servers can perform as authentication and accounting servers or change of authorization (CoA) clients. Authentication servers maintain authentication records for subscribers.

The subscriber daemon requests authentication in RADIUS access-request messages before permitting subscribers access. Accounting servers handle accounting records for subscribers. The subscriber daemon transmits RADIUS accounting-start, interim, and stop messages to the servers. Accounting is the process of tracking subscriber activity and network resource usage in a subscriber session. This includes the session time called time accounting and the number of packets and bytes transmitted during the session called volume accounting. A RADIUS server can behave as a change of authorization (CoA) client, allowing dynamic changes for subscriber sessions. The subscriber daemon supports both RADIUS CoA messages and disconnects messages. CoA messages can modify the characteristics of existing subscriber sessions without loss of service, disconnect messages can terminate subscriber sessions. Each RADIUS request from the subscriber daemon includes the RADIUS accounting-session-id attribute (type 44) with a format that is configurable in the AAA configuration profile and includes at least the subscriber-id to identify the corresponding subscriber. The default format (<subscriber-id>.<timestamp>) includes also a Unix timestamp to ensure that the tuple of NAS-Identifier (e.g. hostname) and Accounting-Session-Id is global and unique to be usable as a key in RADIUS databases.

Additionally, to subscriber-id and accounting-session-id each subscriber consists also of a subscriber-ifl build based on physical port information and subscriber-id (ifp: ifp-0/0/1 and subscriber-id: 72339069014638610 subscriber-ifl: ppp-0/0/1/72339069014638610) which is required as a handle in the RBFS forwarding infrastructure.



```
  Code: Access-Request (1)
  Packet identifier: 0x22 (34)
  Length: 416
  Authenticator: e61a0dd74c74704f608688b08de1dfba
  [The response to this request is in frame 12]
▼ Attribute Value Pairs
  ▶ AVP: t=User-Name(1) l=19 val=user1@rtbrick.com
  ▶ AVP: t=CHAP-Challenge(60) l=18 val=2f696f4e920b47cab869021feb2bf632
  ▶ AVP: t=CHAP-Password(3) l=19 val=02f439040e9feb7bbc9e7622a364344913
  ▶ AVP: t=NAS-IP-Address(4) l=6 val=1.1.1.1
  ▶ AVP: t=NAS-Identifier(32) l=5 val=BNG
  ▶ AVP: t=NAS-Port-Id(87) l=59 val=BNG#hostif-0/0/4#10#7#0.0.0.0/0.0.0.0 eth 1#DEU.RTBRICK.1
  ▶ AVP: t=NAS-Port(5) l=6 val=67149831
  ▶ AVP: t=NAS-Port-Type(61) l=6 val=Ethernet(15)
  ▶ AVP: t=Service-Type(6) l=6 val=Framed(2)
  ▶ AVP: t=Framed-Protocol(7) l=6 val=PPP(1)
  ▶ AVP: t=Acct-Session-Id(44) l=30 val=72339069014638895:1589876315
  ▶ AVP: t=Vendor-Specific(26) l=13 vnd=RtBrick Inc.(50058)
  ▶ AVP: t=Vendor-Specific(26) l=20 vnd=RtBrick Inc.(50058)
  ▶ AVP: t=Vendor-Specific(26) l=16 vnd=RtBrick Inc.(50058)
  ▶ AVP: t=Vendor-Specific(26) l=25 vnd=RtBrick Inc.(50058)
  ▼ AVP: t=Vendor-Specific(26) l=16 vnd=RtBrick Inc.(50058)
      Type: 26
      Length: 16
      Vendor ID: RtBrick Inc. (50058)
    ▶ VSA: t=RtBrick-Subscriber-Id(25) l=10 val=010100000000012f
  ▼ AVP: t=Vendor-Specific(26) l=35 vnd=RtBrick Inc.(50058)
      Type: 26
      Length: 35
      Vendor ID: RtBrick Inc. (50058)
    ▶ VSA: t=RtBrick-Subscriber-Ifl(26) l=29 val=ppp-0/0/4/72339069014638895
  ▶ AVP: t=Vendor-Specific(26) l=29 vnd=The Broadband Forum(3561)
  ▶ AVP: t=Calling-Station-Id(31) l=23 val=0.0.0.0/0.0.0.0 eth 1
  ▶ AVP: t=Vendor-Specific(26) l=21 vnd=The Broadband Forum(3561)
  ▶ AVP: t=Vendor-Specific(26) l=18 vnd=The Broadband Forum(3561)
```

Figure 2. RADIUS Access-Request

> The subscriber-id is an unsigned 64bit integer which is shown as a hex number in Wireshark.

Each subscriber is formed based on configuration profiles and individual settings retrieved via RADIUS. Conflicts between RADIUS-defined attributes and profile attributes are solved by prioritizing those received from RADIUS which is common best practice for broadband access concentrators. New subscribers are signalled via RADIUS access request and either accepted by RADIUS access-accept or rejected by RADIUS access-reject message from the RADIUS server. The RADIUS access-accept includes all attributes required to form the subscriber like IP addresses, DNS servers, and referenced configuration profiles. Some of those attributes can be changed by RADIUS dynamically using CoA requests without disconnecting the subscriber.

RADIUS Accounting

A RADIUS Acct-Status-Type attribute is used by the RADIUS client (subscriber daemon) to mark the start of accounting (for example, upon booting) by specifying Accounting-On and to mark the end of accounting (for example, just before a scheduled reboot) by specifying Accounting-Off. This message is often used by RADIUS servers to automatically close/terminate all open accounting records/sessions for the corresponding client, and therefore must not be sent to servers belonging to a profile that was already used/started for accounting.

Per default, the assumption is that all servers referenced by a RADIUS profile share the same states and therefore accounting-on must be only sent to one of those before the first accounting-start is sent.

RADIUS Accounting-On/Off messages are optionally enabled in the RADIUS profile configuration RADIUS Profile Configuration using the accounting-on-off attribute. The additional attribute accounting-on-wait prevents any new session until accounting has started meaning that the Accounting-On response is received.

> Accounting-Off is currently not implemented!

RADIUS accounting requests are often used for billing and therefore should be able to store and retry over a more extended period (commonly up to 24 hours or more) which can be optionally enabled in the RADIUS profile configuration using the accounting-backup attribute. The maximum backup accounting hold time in seconds is defined in the attribute accounting-backup-max.

RADIUS Redundancy

It is possible to configure multiple RADIUS authentication and accounting servers for redundancy and or load-balancing.

The following two algorithms are supported:

- **DIRECT (default):** Requests are sent to the same server where the last request was sent. If the subscriber daemon receives no response from the server, requests are sent to the next server.
- **ROUND-ROBIN:** Requests are sent to the server following the one where the last request was sent. If the subscriber daemon router receives no response from the server, requests are sent to the next server.

RADIUS NAS-Port-id

The RADIUS attribute NAS-Port-Id (87) is constructed as shown below:

```
<NAS-IDENTIFIER>#<IFP>#<OUTER-VLAN>#<INNER-VLAN>#<ACI>#<ARI>
```

The Agent-Circuit-Id (ACI) and Agent-Remote-Id (ARI) are replaced with an empty string (##) if not available.

PPP over Ethernet (PPPoE)

PPP over Ethernet (PPPoE) is the common standard for internet access in the market.

> ℹ Currently, RBFS only supports PPPoE subscriber sessions with EtherType 0x8100 (802.1Q); it does not support EtherType 0x88a8 (802.1ad).

PPPoE Session-Id

As defined in RFC2516, the PPPoE session-id field is an unsigned 16-bit number with the reserved values 0 for PADI/PADO and 65535 for future use. The session-id will be guaranteed unique per broadcast domain (IFP and VLANs) and client MAC address, but either not unique per device or app instance. The session-id changes every time the session is reconnected.

PPPoE Service-Name

The last service name from the request (PADI or PADR) is internally ignored but copied to the response (PADO or PADS). If the request does not include any service name, the response includes the default service name **access** for compatibility with some clients like Linux pppd.

PPPoE AC-Cookie

This TAG is actually used to aid in protecting against denial-of-service attacks, but it is primarily used in RBFS to decide if a received PADR is a retry for an already answered (PADS send) one. The value itself is unpredictable and generated securely but it does not protect from reply attacks.

If a client receives this TAG in PADO, it MUST return the TAG unmodified in the following PADR. The TAG_VALUE is binary data of any value and length and is not interpreted by the Host.

The AC-Cookie is generated based on 8-bit salt followed by MD5 hash of salt, client MAC and dynamic PPPoE cookie secret.

```
    0                   1                   2
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | SALT          | MD5                           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The PPPoE cookie secret is randomly generated during the PPPoE daemon startup.

The AC-Cookie in the PADR creating the session is stored in the PPPoE PPP session object. For any received PADR it can be checked if there is a session on the same broadcast domain (IFP and VLAN's) and MAC with the same AC-Cookie. In this case, the PADS is just retried.

If the broadcast domain and MAC is equal but AC-Cookie is different, this PADR must be considered as a new request.

This allows to separate two different PPPoE sessions on the same VLAN from the same MAC as frequently used by some service providers.

PPPoE Session Limit

A customer line is typically represented by one (single-tagged) or two VLAN (double-tagged) on a physical interface with a limitation to one session, which is also called the 1:1 VLAN mode.

In some cases, the customer CPE will set up multiple PPPoE sessions on a single VLAN which requires MAC limitations greater than one but less or equal to the per VLAN limitation.

Therefore RBFS supports two different session limitations in the access interface configuration (Access Interface Configuration), one per VLAN (max-subscribers-per-vlan) and an additional per client MAC address (max-subscribers-per-mac) both set to 1 per default as required for 1:1 VLAN mode.

The limitation of sessions per client MAC address must be less or equal the sessions per VLAN and the default set to one for both limits.

PPPoE 1:1 and N:1 Support

RBFS supports both 1:1 VLAN (VLAN Per Subscriber) and N:1 VLAN (shared VLAN) models for subscriber traffic for PPPoE subscribers.

1:1 VLAN (VLAN-per-subscriber) Model

In the 1:1 VLAN model, there is a unique dedicated customer VLAN for a subscriber, that is one VLAN per Subscriber. This model establishes a unique path between each subscriber interface and the router for data transmission by providing traffic separation for every subscriber.

1:1 model operations are relatively simple as it provides one-to-one mapping of specific VLANs to specific subscribers. New services can be added easily without affecting other subscribers and services with this model. However, in a large-scale deployment, this model demands highly scalable and robust routers that can manage many hundreds of VLANs.

The following diagram shows a dedicated customer VLAN for a subscriber, that is, one VLAN per Subscriber.

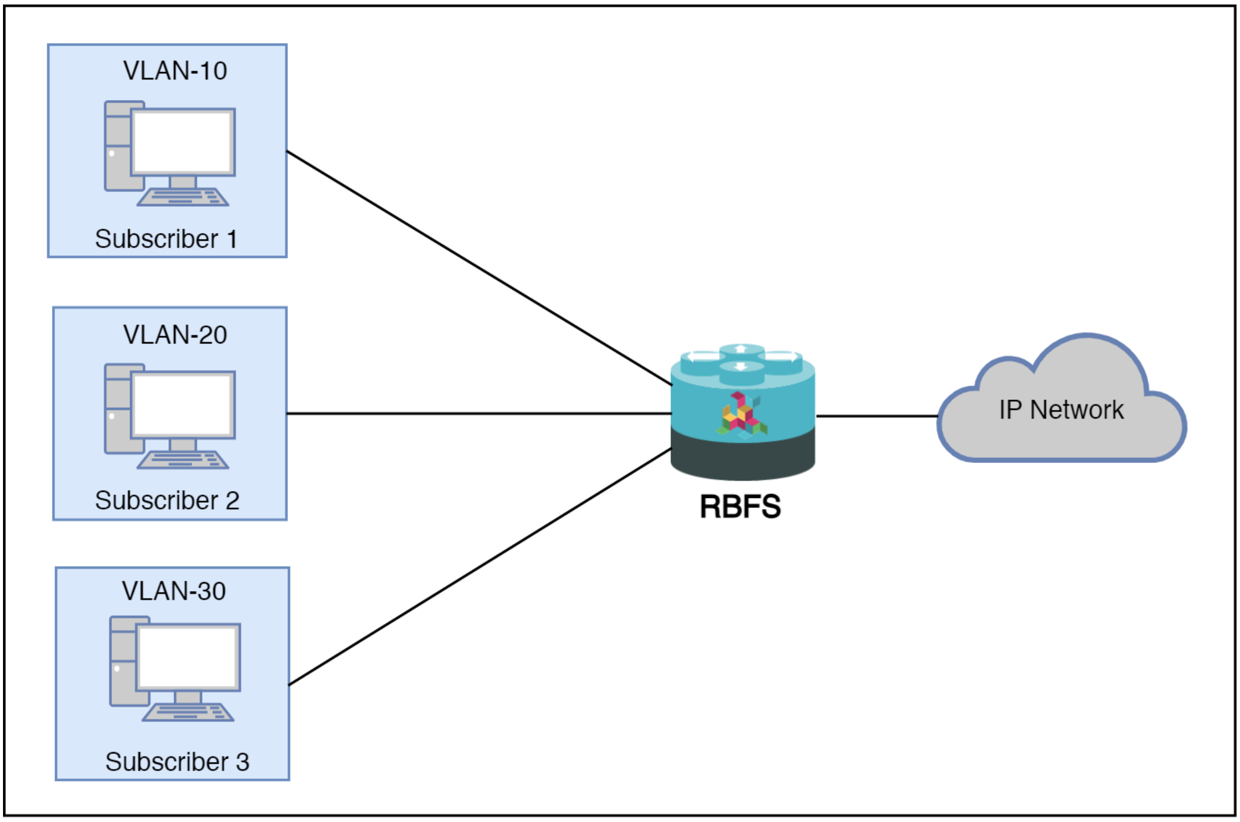


Figure 3. 1:1 VLAN (VLAN-per-subscriber)

N:1 VLAN (Shared VLAN) Model

The Shared VLAN model provides many-to-one (N:1) subscriber-to-service connectivity. This model provides a single VLAN to many subscribers. Unlike in the 1:1 model, in which the VLAN is dedicated to a customer, N:1 provides a shared VLAN to many subscribers, and this VLAN carries all types of service (i.e., data, voice, video, etc.). One disadvantage of shared VLAN is the lack of logical isolation between user sessions at the VLAN level.

The following diagram shows a single VLAN that is connected to many subscribers.

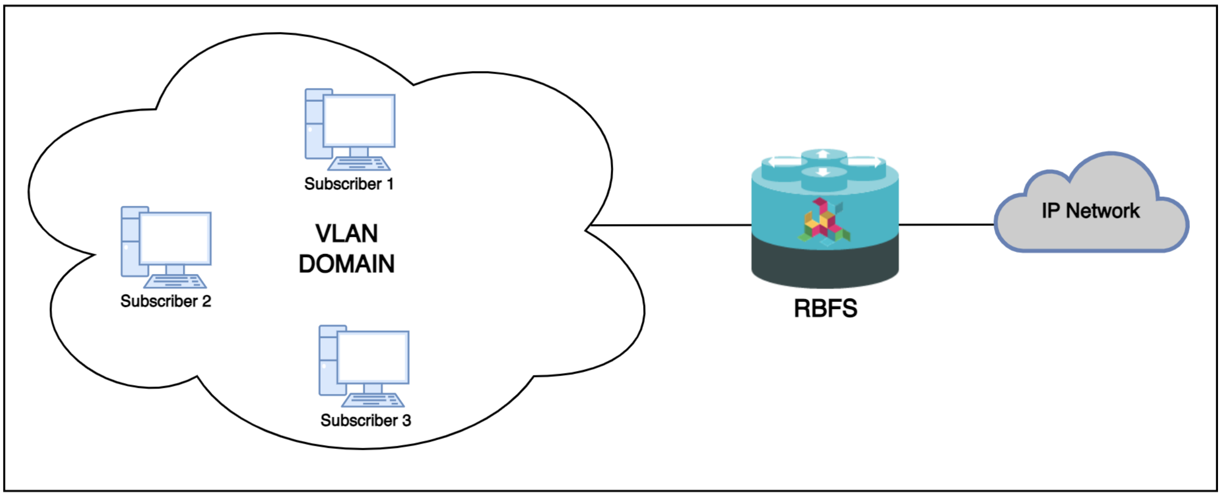


Figure 4. N:1 VLAN (Shared VLAN-per-service)

PPPoE MTU Profiles

The PPP protocol allows each endpoint to negotiate a maximum receive unit (MRU). This MRU is applied as the maximum transmission unit (MTU) on the other end of the PPP connection. Each endpoint negotiates its own MRU with the other peer. Thus using a different MTU per direction is not uncommon for PPP, even if not desired.

Bare metal switch hardware is typically limited in the number of supported MTU values. So RBFS has introduced the concept of MTU profiles with different types like physical, ip or pppoe. The last one is reserved for use with PPPoE sessions and applies to IPv4 and IPv6 traffic.

```
supervisor@switch: cfg> show config forwarding-options mtu-profile
{
  "rtbrick-config:mtu-profile": [
    {
      "mtu-profile-name": "IP-MTU-1500",
      "size": 1500,
      "type": "ip",
      "action": "redirect-to-cpu"
    },
    {
      "mtu-profile-name": "IP-MTU-9202",
      "size": 9202,
      "type": "ip",
      "action": "drop"
    },
    {
      "mtu-profile-name": "PPPoE-MTU-1320",
      "size": 1320,
      "type": "pppoe",
      "action": "redirect-to-cpu"
    },
    {
      "mtu-profile-name": "PPPoE-MTU-1456",
      "size": 1456,
      "type": "pppoe",
      "action": "redirect-to-cpu"
    },
    {
      "mtu-profile-name": "PPPoE-MTU-1472",
      "size": 1472,
      "type": "pppoe",
      "action": "redirect-to-cpu"
    },
    {
      "mtu-profile-name": "Port-MTU-9216",
      "size": 9216,
      "type": "physical",
      "action": "drop"
    },
    {
      "mtu-profile-name": "__default_pppoe__",
      "size": 1492,
      "type": "pppoe",
      "action": "redirect-to-cpu"
    }
  ]
}
```

A configured size of 1492 bytes limits the size of the IPv4 or IPv6 header plus payload.

> The physical access interface should be configured with an MTU profile large enough to serve all PPPoE MTU profiles, including the additional overhead for PPPoE and VLAN headers. Further details about interface MTU profiles can be found in the *Interfaces Configuration Guide*.

The action could be either drop or redirect-to-cpu. The action drop silently discards all oversized packets. The action redirect-to-cpu punts oversized packets to the CPU where those could be either fragmented or dropped with ICMP response.

The Q2C platform supports up to 8 MTU profiles in total. The amount of pppoe profiles is limited to 6 including the default profile *_default_pppoe_*. The default profile can’t be deleted but overwritten to change size and action.

```
supervisor@switch: cfg> show config forwarding-options mtu-profile
__default_pppoe__
{
  "rtbrick-config:mtu-profile": [
    {
      "mtu-profile-name": "__default_pppoe__",
      "size": 1492,
      "type": "pppoe",
      "action": "redirect-to-cpu"
    }
  ]
}
```

The command show pppoe mtu-profile lists all PPPoE MTU profiles in increasing order, with two statistics about the exact and best match.

```
supervisor@switch: op> show pppoe mtu-profile
Profile                MTU      Exact Match       Best Match
PPPoE-MTU-1320         1320     0                 0
PPPoE-MTU-1456         1456     0                 0
PPPoE-MTU-1472         1472     0                 0
__default_pppoe__      1492     0                 0
```

If a client requests an MRU via PPP LCP Configure-Request, this value is used to search for an appropriate MTU profile. This is done by iterating over the ordered list of MTU profiles, as long as the received MRU is greater than the MTU size. If the requested MRU is found in the list of MTU profiles, the exact match counter is incremented, and the MRU is accepted.

In case the requested MRU is not found, the last MTU profile found is selected and the best match counter is incremented. The selected MTU is then offered to the client via PPP LCP Configure-Nak. If the offered MRU is not accepted by the client after three offers, a fallback profile is applied. This means that the requested MRU from the client is accepted, but the largest pppoe MTU profile is applied. This algorithm was built to ensure most client compatibility.

Let’s assume a client requests the MRU 1482, but only profiles for 1472 and 1492 are configured. In this case, 1472 is offered as the best match via PPP LCP Configure-Nak. The client could either accept the offer by sending a PPP LCP Configure-Request with the MRU 1472 or try again with the original value of 1482. This is repeated up to three times before the fallback profile is applied. In this case, the client MRU of 1482 is accepted but the maximum MTU of 1492 is applied. The majority of CPE devices support TCP MSS clamping using the negotiated MRU of 1482. So at least TCP traffic is still limited to the negotiated MRU. This is the reason for applying a larger MTU profile as the fallback profile. It's also common that a CPE still accepts packets larger than the negotiated MRU.

The exact and best match counters can be used by operators to verify if the configured MTU profiles fit their environment or should be adopted.

The negotiated MRU and applied MTU can be verified with the following command for every single PPPoE session.

```
user@switch: op> show pppoe session 72339069014638648 detail
Subscriber-Id: 72339069014638648
    State: ESTABLISHED
    ...
    PPP LCP:
        ...
        MRU: 1492 Peer: 1492
        MTU: 1492 Profile: __default_pppoe__
    ...
```

For L2TPv2 tunneled PPPoE sessions, the MTU is enforced by the LNS. It's usual behavior that the LNS renegotiates the MTU. So LAC may not know the actual MTU. This is the reason why RBFS does not apply an MTU profile for such sessions.

RBFS overwrites the selected MTU profile with *default_l2tp* for L2TPv2 sessions. This profile must be explicitly created, otherwise it is ignored. The action must be drop because ICMP or fragmentation is not supported for tunneled sessions.

```
supervisor@switch: cfg> show config forwarding-options mtu-profile
__default_l2tp__
{
    "rtbrick-config:mtu-profile": [
      {
        "mtu-profile-name": "__default_l2tp__",
        "size": 1492,
        "type": "pppoe",
        "action": "redirect-to-cpu"
      }
    ]
  }
```

This configuration allows to optionally enforce an MTU on LAC if needed.

PPPoE VLAN Profiles

This chapter describes the VLAN profile feature. If enabled for the access interface, then incoming sessions (e.g. PPPoE PADI/PADR) are not honored unless matching vlan-profile is found.

The VLAN profiles must be added to the table global.vlan.profile owned by PPPoE daemon. All entries in this table are ephemeral and therefore lost after reboot or PPPoE daemon restart.

Example:

```
{
    "table": {
        "table_name": "global.vlan.profile"
    },
    "objects": [
        {
            "attribute": {
                "ifp_name": "ifp-0/1/2",
                "outer_vlan_min": 128,
                "outer_vlan_max": 128,
                "inner_vlan_min": 1,
                "inner_vlan_max": 4095,
                "access_profile_name": "access-profile-vlan"
            }
        }
    ]
}
```

PPPoE Dual-Stack IPv4/IPv6 with DHCPv6

The whole IPv6 control plane of a PPPoE session like ICMPv6 router-solicitation (RS), ICMPv6 router-advertisement (RA) and DHCPv6 is handled in the PPPoE daemon (pppoed).

The PPPoE daemon handles received router solicitations by responding with router advertisements and is sending frequent router advertisements based on configured intervals.

The other-config flag in the router-advertisement is automatically set if DHCPv6 is enabled for this particular subscriber. This flag signals that there is more information available via DHCPv6.

DHCPv6 over PPPoE is different to DHCPv6 over Ethernet because of the special characteristics of point-to-point protocols. DHCPv6 over PPPoE is supporting delegated IPv6 prefixes (IA_PD) and DNS options only. Unsupported IA options (IA_NA and IA_TA) or options that can be served will be rejected with status code options as defined per RFC.

The delegated IPv6 prefix served by DHCPv6 will be assigned to the subscriber via RADIUS or local pool regardless of the protocols negotiated with the client. DHCPv6 was primarily designed for use in Ethernet networks. The fact that Ethernet is connectionless requires that DHCPv6 servers must manage releases for the clients and free them automatically if a lease expires. Such extensive release management is not needed for connection-oriented protocols like PPPoE where addresses are assigned to the PPPoE session. This fact allows to implementing DHCPv6 nearly stateless on the server side by just tracking if an assigned prefix is assigned or released. This is tracked in the attribute ipv6pd_negotiated of the PPPoED/SubscriberD (global.ppp.1.subscriber.result) result object and copied to the actual subscriber object (local.access.subscriber). As this use case is covered by PPPoE, there is no lease expiry implemented.

The delegated-prefix is added to the subscriber-ifl only if negotiated and removed if not negotiated. The presence of delegated prefix in the subscriber-ifl is used by IFMD to add or remove the forwarding entry.

If DHCPv6 is enabled but no delegated prefix provided, only DNS is served in response if available.

PPPoE DHCPv6 Server DUID

The DHCPv6 server identifier DUID is generated based on IP6CP negotiated interface-identifier as shown below:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     DUID-Type 3 (DUID-LL)     |    hardware type 27 (EUI64)   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     interface-identifier                      |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Layer Two Tunneling Protocol (L2TPv2)

This chapter describes the RtBrick Layer Two Tunneling Protocol (L2TPv2) implementation. This document describes also the corresponding configuration and operations commands for PPPoE access services with PPP tunneling using the Layer Two Tunneling Protocol version 2 (L2TPv2) on RtBRick FullStack (RBFS).

Typically, a user obtains a Layer 2 (L2) point-to-point connection to a Broadband Network Gateway (BNG) using the PPPoE protocol as described in RFC 2516 and runs PPP over that connection. In the most common case, the L2 termination point and PPP session endpoint reside on the same physical device. Tunneling protocols, such as L2TPv2 provide a dynamic mechanism for extending PPP by allowing the L2 and PPP endpoints reside on different devices that are interconnected by an IP network. This separation allows the actual processing of PPP packets to be divorced from the termination of the L2 circuit. The L2TP access concentrator (LAC) physically terminates the L2 connection and tunnels the PPP packets across an IP network to the L2TP network server (LNS). The LNS then terminates the logical PPP connection.

RFC and draft compliance are partial except as specified.

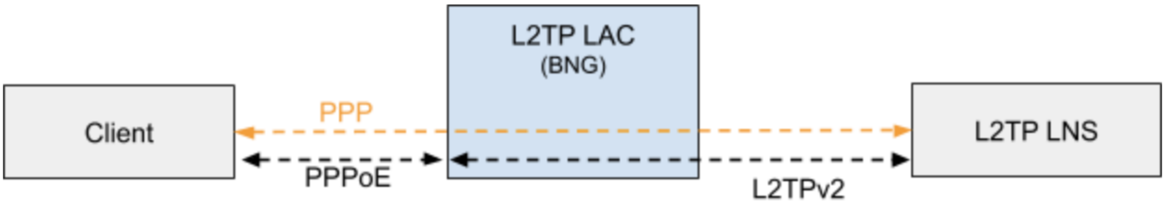


Figure 5. L2TP PPPoE

To establish a PPPoE session via L2TP, the tunnel-type must be configured as L2TP. This configuration can be achieved either for local users or by utilizing the corresponding tunnel-type attribute through RADIUS.

```
# Local User
cfg> set access user-profile local@l2tp tunnel-type L2TP

# FreeRADIUS
"radius@l2tp" Cleartext-Password := "test"
    Service-Type = Framed-User,
    Framed-Protocol = PPP,
    Tunnel-Type:0 = L2TP
```

Defining an L2TP configuration profile is essential, which can be referenced through an access-profile or by employing the appropriate RADIUS VSA. The actual tunnels may either be defined locally via an L2TP pool configuration or set up dynamically through RADIUS.

L2TP LAC

The L2TP Access Concentrator (LAC) is a node that acts as one side of an L2TP tunnel endpoint, and is a peer to the L2TP Network Server L2TP LNS. The LAC sits between a LNS and a remote system, and forwards packets to and from each.

L2TP LNS

The L2TP Network Server (LNS) is a node that acts as one side of an L2TP tunnel endpoint and is a peer to the L2TP Access Concentrator L2TP LAC. The LNS is the logical termination point of a PPP session that is being tunneled from the remote system by the LAC.

> The LNS role is currently not supported!

L2TP Tunnel Selection

Each new session creates a session request object (local.l2tp.session.request) to track the tunnel selection progress, the currently selected ones, and which are already tried. This object is automatically deleted if the session setup is successful.

All tunnels in state DEAD are skipped in the tunnel selection but considered at the end if no other tunnels are available. Tunnels with a session limit reached are not considered for further sessions. To select a tunnel, the L2TP daemon first generates list of preferred tunnels based on tunnel preference, where the lowest value has the highest priority. The configured L2TP tunnel selection algorithm decides how to select a tunnel out of the remaining tunnels with the same preference. The RANDOM algorithm selects the tunnel randomly whereas BALANCED selects the least filled tunnel based on the number of sessions.

Following the L2TP tunnel pool order/priority in case, there are multiple pools available for a single subscriber:

1. RADIUS defined tunnel (RFC2866)
2. RADIUS VSA (RtBrick-L2TP-Pool) or local user profile
3. L2TP configuration profile

L2TP Control Channel

The control channel is responsible for the orderly passing of control messages between the tunnel endpoints and acts as a transport layer for reliable delivery of control messages and tunnel keep alive services for the tunnel.

Each L2TP tunnel is split into the actual tunnel object with all the information exchanged during tunnel establishment plus the FSM state and a separate control channel with the sequence numbers, window size, and thresholds changed with every sent and received packet.

RBFS sent a ZLB ACK only if there are no further messages waiting in queue for that peer, as well as to acknowledge multiple packets at once.

The HELLO keep-alive messages are also part of the control channel and only send if there is no other message sent if the queue is empty and no other message send during the hello interval.

L2TP Access Line Information (RFC5515)

Connect-Speed-Update-Notification (CSUN)

The Connect-Speed-Update-Notification (CSUN) is an L2TP control message sent by the LAC to the LNS to provide transmit and receive connection speed updates for one or more sessions.

This implementation will send one CSUN request per session!

CSUN requests are disabled per default and can be enabled in the L2TP profile L2TP Profile Configuration.

CSUN messages are defined in RFC5515, which is not widely supported. Therefore those messages are marked as not mandatory in RBFS to allow interwork with LNS servers not supporting RFC5515.

RFC2661:

```
The Mandatory (M) bit within the Message Type AVP has special
meaning. Rather than an indication of whether the AVP itself
should be ignored if not recognized, it is an indication as to
whether the control message itself should be ignored. Thus, if the
M-bit is set within the Message Type AVP and the Message Type is
unknown to the implementation, the tunnel MUST be cleared.  If the
M-bit is not set, then the implementation may ignore an unknown
message type.
```

RFC and draft compliance are partial except as specified.

Connect-Speed-Update-Request (CSURQ)

The Connect-Speed-Update-Request (CSURQ) is an L2TP control message sent by the LNS to the LAC to request the current transmission and receive connection speed for one or more sessions.

> Sending or responding to CSURQ requests is currently not supported!

Access Line Information L2TP Attribute Value Pair Extensions

The corresponding access line information for a subscriber is included in the ICRQ message as defined in RFC5515.

Connect Speed Values

The default value for TX and RX Connect Speed is set to 1000000000 (1G) which is replaced by the actual data rate upstream/downstream of the corresponding access line information object or directly set using the RADIUS attributes RtBrick-L2TP-Tx-Connect-Speed (42) and RtBrick-L2TP-Rx-Connect-Speed (43).

IPoE

IP-over-Ethernet (IPoE) is a popular alternative to PPPoE based access using DHCP for IPv4 and DHCPv6 for IPv6 where both protocols are handled in the IPoE daemon (ipoed).

IPoE subscribers are identified by IFP, optional VLAN and client MAC address.

The dynamic creation of IPoE subscribers is triggered by the first DHCPv4 discovery or DHCPv6 solicit request received. Any response is postponed until the subscriber is successfully authenticated using known authentication methods like none, local or RADIUS. For DHCP mode server all addresses are assigned during authentication to the subscriber and used by DHCPv4 and DHCPv6 to handle client requests. For the DHCP relay mode, all IP addresses are allocated by an external DHCP server.

IPoE subscribers will be terminated automatically if all protocol bindings are deleted.

IPoE 1:1 VLAN Support

RBFS supports 1:1 VLAN (VLAN Per Subscriber) model for subscriber traffic for IPoE subscribers. In the 1:1 VLAN model, there is a unique dedicated customer VLAN for a subscriber, that is one VLAN per Subscriber. This model establishes a unique path between each subscriber interface and the router for data transmission by providing traffic separation for every subscriber.

The following diagram shows a dedicated customer VLAN for a subscriber, that is, one VLAN per Subscriber.

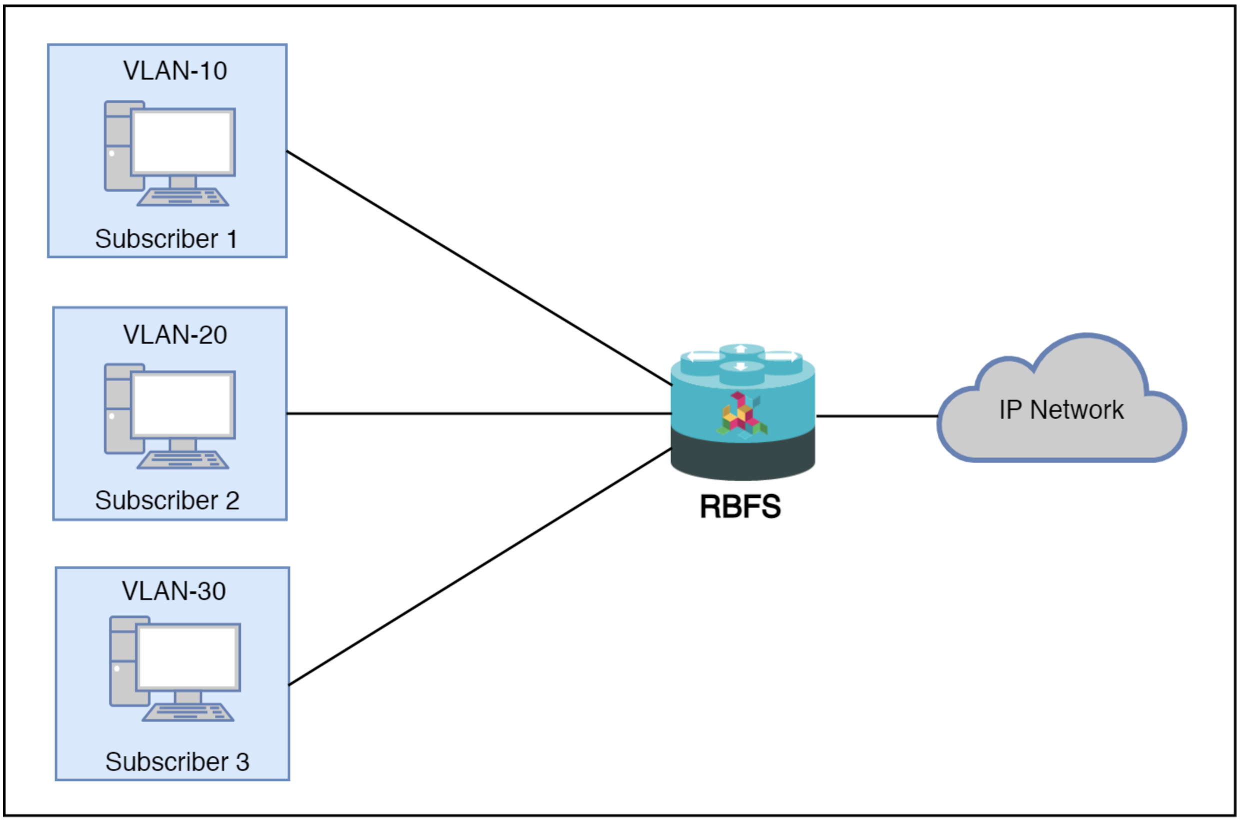


Figure 6. 1:1 VLAN (VLAN-per-subscriber)

IPoE Session Limit

A customer line is typically represented by one (single-tagged) or two VLAN (double-tagged) on a physical interface with a limitation to one subscriber, which is also called the 1:1 VLAN mode.

IPoE subscribers are implicitly limited to max one per MAC address, as client MAC address is used as part of the key to identify subscribers.

IPoE Username

For each IPoE subscriber, a username is generated automatically using the client’s MAC address followed by @ipoe.

Example: fe:08:e8:ea:1d:32@ipoe

IPoE DHCPv6 Server DUID

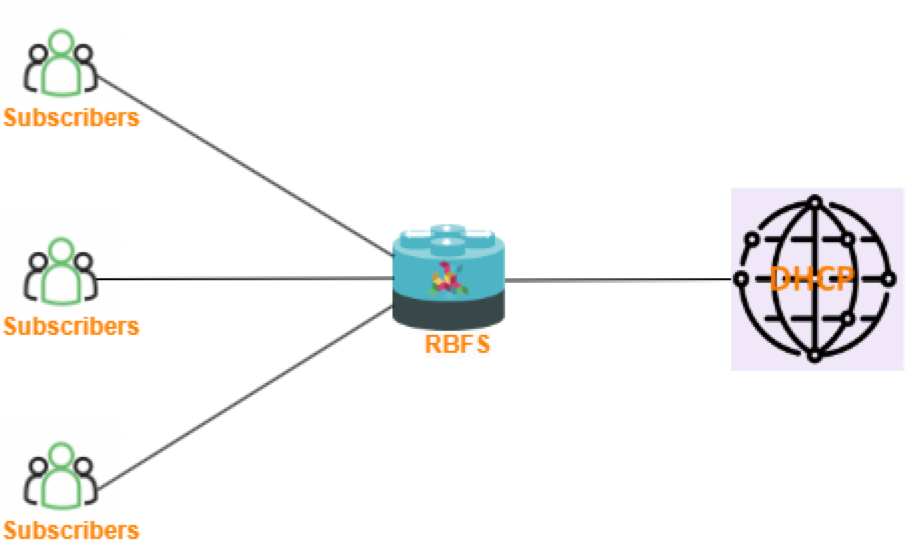
The generated DHCPv6 server identifier DUID is from type 3 (DUID-LL) with hardware type 27 (EUI64) and IFP MAC address to derive the EUI64 interface identifier.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     DUID-Type 3 (DUID-LL)     |    hardware type 27 (EUI64)   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                      interface-identifier                     |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IPoE DHCP Relay

When IPoE is configured on relay mode, the system relies on an external DHCP server for IP address allocation and client configuration functions. The DHCP server contains a pool of IP addresses, and from that pool, it allocates addresses to subscribers. RBFS acts as a DHCP relay during its interaction with the DHCP server. The feature allows multiple RBFS instances to use a single centralized DHCP server for their IP allocation.

For information on how to configure IPoE in relay mode, see the section DHCPv4.



4.1.2. Subscriber Management Configuration

The following sections describe Subscriber Management configuration syntax and commands.

Configuration Hierarchy

The main interface configuration for a physical interface (ifp) and associated VLANs is related to a series of profiles that hold parameters for authentication with AAA, services like IGMP and MLD, access methods like PPPoE and the like, and so on. The overall structure of this configuration and profile system is shown in Figure 2.

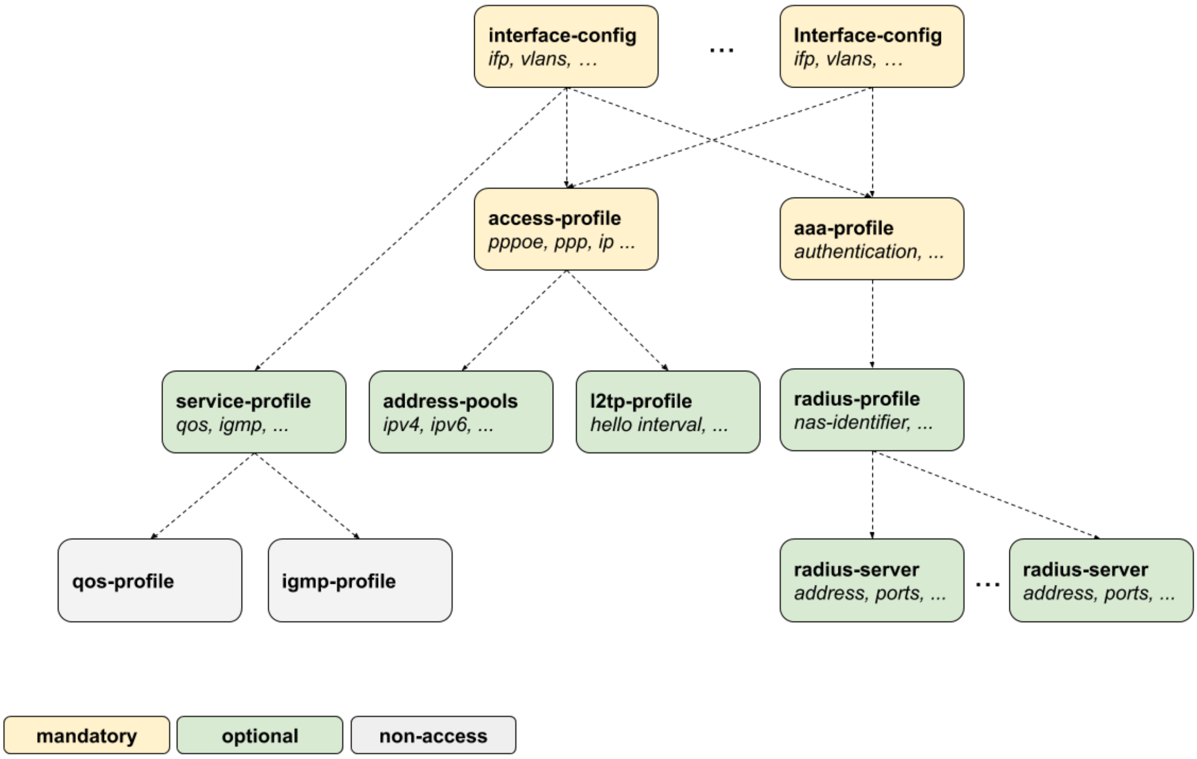


Figure 7. Configuration and Profiles

All of the access configuration and profile sections are edited under the **access** top-level hierarchy of the configuration.

```
supervisor@switch: cfg> set access
  <cr>
  aaa-profile            Global AAA profile configuration
  access-profile         Global access profile configuration
  chassis-id             Chassis id for this node <0-15>
  dhcp-relay             Global DHCP relay configuration
  dhcp-server            Global DHCP server configuration
  dhcpv6-server          Global DHCP server configuration
  interface              Global interface profile configuration
  l2bsa                  Global access l2bsa configurations
  l2tp-pool              Global L2TPv2 pool configuration
  l2tp-profile           Global L2TPv2 profile configuration
  pool                   Global address pool configuration
  radius-profile         Global AAA RADIUS profile configuration
  radius-server          Global RADIUS server configuration
  service-profile        Global service profile configuration
  user-profile           Global user profile configuration
```

Detailed descriptions of each configuration and profile can be found in the following chapters. This configuration guide starts with the interface configuration which is the entry point for every new subscriber followed by mandatory access and AAA configuration profiles.

- **interface-config** Access Interface Configuration
- **access-profile** Access Profile Configuration
- **aaa-profile** AAA Profile Configuration

The second part explains the optional configurations.

- **radius-profile** RADIUS Profile Configuration
- **radius-server** RADIUS Server Configuration
- **service-profile** Service Profile Configuration
- **l2tp-profile** L2TP Profile Configuration
- **address-pools** Address Pool Configuration

The user-profile and l2tp-pool are the only components not referenced by name. The key here is the user or pool name.

- **user-profile** User Profile Configuration
- **l2tp-pool** L2TP Tunnel Pool Configuration

Access Interface Configuration

Table: global.access.interface.config

Although there is no correct way to configure subscriber management, it makes most sense to proceed from mandatory configurations and profiles to optional ones. First and foremost, among these mandatory configuration items is the access interface configuration which is the anchor point for almost all further access configurations.

The interface configuration assigns the access type, access profile Access Profile Configuration, AAA profile AAA Profile Configuration

Multiple interface configurations per IFP with disjoint VLAN ranges are supported.

The way that the interface configuration relates to all subscriber management configuration tasks is shown in the picture below.

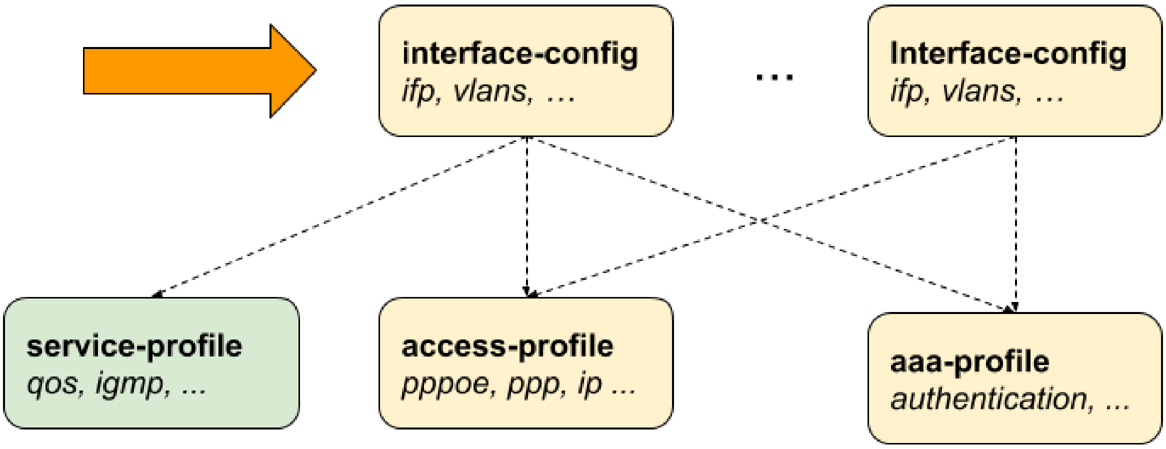


Figure 8. Access Interface Configuration

Note that there can be more than one interface configured for subscriber management and each interface can reference the same profiles.

There are four major configuration tasks for the access interface:

1. Configure the physical interface name (IFP) and VLAN range
2. Configure the mandatory access type (PPPoE or IPoE)
3. Configure the mandatory access profile
4. Configure the mandatory AAA profile
5. Configure optional attributes like service profile or session limit

Configuring Access Interfaces

Access interfaces can be configured without VLAN tags (untagged) and with one (single tagged) or two (double tagged) VLAN tags.

```
supervisor@switch: cfg> set access interface
  <cr>
  double-tagged         Double tagged access
  single-tagged         Single tagged access
  untagged              Untagged access

supervisor@switch: cfg> set access interface untagged ifp-0/0/0
  <cr>
  aaa-profile-name          AAA profile name
  access-profile-name       Access profile name
  access-type               Access service type
  max-subscribers-per-mac   Restrict maximum subscribers per MAC address
  max-subscribers-per-vlan  Restrict maximum subscribers per VLAN
  service-profile-name      Service profile name
  vlan-profile-enable       Enable VLAN profiles
```

The following example shows an untagged access interface.

```
supervisor@switch: cfg> show config access interface untagged ifp-0/0/0
{
  "rtbrick-config:untagged": {
    "interface-name": "ifp-0/0/0",
    "access-type": "PPPoE",
    "access-profile-name": "pppoe-dual",
    "service-profile-name": "service-profile1",
    "aaa-profile-name": "aaa-radius",
    "vlan-profile-enable": "true",
    "max-subscribers-per-vlan": 1,
    "max-subscribers-per-mac": 1
  }
}
```

Attribute	Description
access-type	The mandatory access type attribute define the access protocol used for this interface. **Values:** PPPoE or IPoE
access-profile-name	The name of the mandatory access profile Access Profile Configuration.
aaa-profile-name	The name of the mandatory AAA profile AAA Profile Configuration.
service-profile-name	This option allows assigning an optional service profile Service Profile Configuration which can be dynamically overwritten via RADIUS.
max-subscribers-per-vlan	This option defines the maximum number of subscribers per IFP and VLAN. A value of 1 will implicitly set the VLAN mode to 1:1, where any value grater than 1 means N:1. **Default:** 1 **Range:** 1 - 65535 ℹ There is currently no support for more than one PPPoE session or IPoE subscriber per VLAN for Broadcom QMX (Qumran)!

Attribute	Description
max-subscribers-per-mac	Maximum number of subscribers per IFP, VLAN, and MAC. This option must be less or equal to the max-subscribers-per-vlan. **Default:** 1 **Range:** 1 - 65535
vlan-profile-enable	If enabled, incoming PPPoE sessions (PPPoE PADI/PADR) are not honored unless matching vlan-profile is found in the table global.vlan.profile of the PPPoE daemon. VLAN profiles are described in detail in PPPoE VLAN Profiles. **Default:** false
gateway-ifl	This options selects the IPoE gateway IFL (unnumbered source IFL) which is typically a loopback interface used as a gateway for IPoE subscribers.

Configuring Untagged Interfaces

```
supervisor@switch: cfg> set access interface untagged
  <interface-name>      Name of the physical interface

supervisor@switch: cfg> set access interface untagged ifp-0/0/0
  <cr>
  aaa-profile-name          AAA profile name
  access-profile-name       Access profile name
  access-type               Access service type
  max-subscribers-per-mac   Restrict maximum subscribers per MAC address
  max-subscribers-per-vlan  Restrict maximum subscribers per VLAN
  service-profile-name      Service profile name
  vlan-profile-enable       Enable VLAN profiles

supervisor@switch: cfg> set access interface untagged ifp-0/0/0 access-type PPPoE
supervisor@switch: cfg> set access interface untagged ifp-0/0/0 access-profile-
name pppoe-dual
supervisor@switch: cfg> set access interface untagged ifp-0/0/0 aaa-profile-name
aaa-radius
supervisor@switch: cfg> commit
supervisor@switch: cfg> show config access interface untagged ifp-0/0/0
{
  "rtbrick-config:untagged": {
    "interface-name": "ifp-0/0/0",
    "access-type": "PPPoE",
    "access-profile-name": "pppoe-dual",
    "aaa-profile-name": "aaa-radius"
  }
}
```

Untagged interfaces are not supported on Broadcom QMX and

> QAX platforms.

Configuring Single VLAN Tagged Interfaces

The VLAN range 128 - 4000 includes VLAN 128, 4000, and VLAN identifiers between.

```
supervisor@switch: cfg> set access interface single-tagged
  <interface-name>      Name of the physical interface

supervisor@switch: cfg> set access interface single-tagged ifp-0/0/0
  <outer-vlan-min>      Outer VLAN min

supervisor@switch: cfg> set access interface single-tagged ifp-0/0/0 128
  <outer-vlan-max>      Outer VLAN max

supervisor@switch: cfg> set access interface single-tagged ifp-0/0/0 128 3000
  <cr>
  aaa-profile-name          AAA profile name
  access-profile-name       Access profile name
  access-type               Access service type
  max-subscribers-per-mac   Restrict maximum subscribers per MAC address
  max-subscribers-per-vlan  Restrict maximum subscribers per VLAN
  service-profile-name      Service profile name
  vlan-profile-enable       Enable VLAN profiles

supervisor@switch: cfg> set access interface single-tagged ifp-0/0/0 128 3000
access-type PPPoE
supervisor@switch: cfg> set access interface single-tagged ifp-0/0/0 128 3000
access-profile-name pppoe-dual
supervisor@switch: cfg> set access interface single-tagged ifp-0/0/0 128 3000 aaa-
profile-name aaa-radius
supervisor@switch: cfg> commit
supervisor@switch: cfg> show config access interface single-tagged ifp-0/0/0 128
3000
{
  "rtbrick-config:single-tagged": [
    {
      "interface-name": "ifp-0/0/0",
      "outer-vlan-min": 128,
      "outer-vlan-max": 3000,
      "access-type": "PPPoE",
      "access-profile-name": "pppoe-dual",
      "aaa-profile-name": "aaa-radius"
    }
  ]
}
```

Configuring Double VLAN Tagged Interfaces

Configuring the minimum and maximum VLAN settings to an identical value results in achieving an exact match.

Currently RBFS only supports PPPoE subscriber sessions with EtherType 0x8100 (802.1Q); it does not support EtherType 0x88a8 (802.1ad).

```
supervisor@switch: cfg> set access interface double-tagged
  <interface-name>      Name of the physical interface

supervisor@switch: cfg> set access interface double-tagged ifp-0/0/0
  <outer-vlan-min>      Outer VLAN min

supervisor@switch: cfg> set access interface double-tagged ifp-0/0/0 128
  <outer-vlan-max>      Outer VLAN max

supervisor@switch: cfg> set access interface double-tagged ifp-0/0/0 128 3000
  <inner-vlan-min>      Inner VLAN min

supervisor@switch: cfg> set access interface double-tagged ifp-0/0/0 128 3000 7
  <inner-vlan-max>      Inner VLAN max

supervisor@switch: cfg> set access interface double-tagged ifp-0/0/0 128 3000 7 7
  <cr>
  aaa-profile-name          AAA profile name
  access-profile-name       Access profile name
  access-type               Access service type
  max-subscribers-per-mac   Restrict maximum subscribers per MAC address
  max-subscribers-per-vlan  Restrict maximum subscribers per VLAN
  service-profile-name      Service profile name
  vlan-profile-enable       Enable VLAN profiles

supervisor@switch: cfg> set access interface double-tagged ifp-0/0/0 128 3000 7 7
access-type PPPoE
supervisor@switch: cfg> set access interface double-tagged ifp-0/0/0 128 3000 7 7
access-profile-name pppoe-dual
supervisor@switch: cfg> set access interface double-tagged ifp-0/0/0 128 3000 7 7
aaa-profile-name aaa-radius
supervisor@switch: cfg> commit
supervisor@switch: cfg> show config access interface single-tagged ifp-0/0/0 128
3000 7 7
{
  "rtbrick-config:double-tagged": {
    "interface-name": "ifp-0/0/0",
    "outer-vlan-min": 128,
    "outer-vlan-max": 3000,
    "inner-vlan-min": 7,
    "inner-vlan-max": 7,
    "access-type": "PPPoE",
    "access-profile-name": "pppoe-dual",
    "aaa-profile-name": "aaa-radius"
  }
}
```

Access Profile Configuration

While it is mandatory to configure an interface with an access profile name, such as pppoe-dual, it is still necessary to configure the properties and parameters of the access profile itself.

The picture below shows how the access profile configuration is related to all subscriber management configuration tasks.

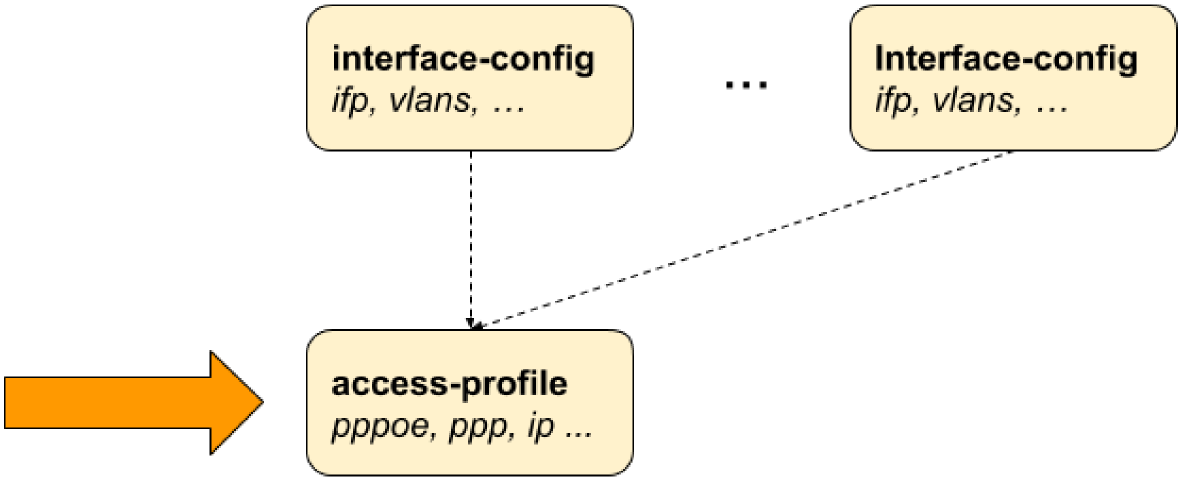


Figure 9. Access Profile Configuration

Configuring the Access Profile

```
supervisor@switch: cfg> set access access-profile
  <profile-name>        Name of the access profile

supervisor@switch: cfg> set access access-profile pppoe-dual
  <cr>
  address-family        Address-family configuration
  instance              Instance name
  protocol              Protocol configuration
```

Attribute	Description
instance	Change routing instance. **Default:** default

The following examples show typical access profiles for PPPoE and IPoE with IPv4 and IPv6.

PPPoE with IPv4 and IPv6:

```
supervisor@switch: cfg> show config access access-profile pppoe-dual
{
  "rtbrick-config:access-profile": {
    "profile-name": "pppoe-dual",
    "instance": "default",
    "protocol": {
      "pppoe": {
        "enable": "true",
        "session-protection": {
          "enable": "true"
        },
        "vlan-priority": 6
      },
      "ppp": {
        "lcp": {
          "authentication-protocol": "PAP_CHAP",
          "echo-interval": 30,
          "echo-max-retransmit": 3,
          "echo-enable": "true"
        },
        "ipcp": {
          "enable": "true",
          "source-ifl": "lo-0/0/0/1"
        },
        "ip6cp": {
          "enable": "true"
        }
      },
      "ra": {
        "enable": "true",
        "interval": 60
      },
      "dhcpv6": {
        "enable": "true"
      },
      "l2tp": {
        "tunnel-profile": "l2tp-default"
      }
    },
    "address-family": {
      "ipv4": {
        "enable": "true",
        "primary-dns": "198.51.100.1",
        "secondary-dns": "198.51.100.4"
      },
      "ipv6": {
        "enable": "true",
        "primary-dns": "2001:db8:0:100::",
        "secondary-dns": "2001:db8:0:104::"
      }
    }
  }
}
```

IPoE with IPv4 and IPv6:

```
supervisor@switch: cfg> show config access access-profile ipoe-dual
{
  "rtbrick-config:access-profile":{
    "profile-name":"ipoe",
    "protocol":{
      "dhcp":{
        "enable":"true",
        "mode":"server"
      },
      "dhcpv6":{
        "enable":"true",
        "mode":"server"
      }
    },
    "address-family":{
      "ipv4":{
        "enable":"true",
        "proxy-arp-enable": "true",
        "pool-name":"ipoe",
        "primary-dns":"198.51.100.1,
        "secondary-dns":"198.51.100.4"
      },
      "ipv6":{
        "enable":"true",
        "pool-name":"ipoe-ia-na",
        "prefix-delegation-pool-name":"ipoe-ia-pd",
        "primary-dns": "2001:db8:0:100::",
        "secondary-dns": "2001:db8:0:104::"
      }
    }
  }
}
```

Configuring IPv4

The address family IPv4 must be explicitly enabled in the access profile to be available for access protocols like PPP (PPPoE) or DHCP (IPoE).

```
supervisor@switch: cfg> set access access-profile pppoe-dual address-family ipv4
  <cr>
  enable                Enable IPv4
  pool-name             Local IPv4 pool name
  primary-dns           Primary DNS server
  proxy-arp-enable      Enable Proxy ARP
  secondary-dns         Secondary DNS server
  static-ipv4           Static address
  dad-enable            Enable/disable IPv4 duplicate address detection (Enabled
by default)
```

Attribute	Description
enable	Enable IPv4 **Default:** false
pool-name	The optional pool-name attribute allows assigning the IPv4 address from a local managed pool as described in Address Pool Configuration. This address is used by protocols like PPP IPCP (PPPoE) or DHCP (IPoE) as a client or peer IPv4 address.

Attribute	Description
primary-dns secondary-dns	The primary-dns and secondary-dns servers configured are used by protocols like PPP (PPPoE) or DHCP (IPoE) and advertised to the client.
proxy-arp-enable	Enable/disable proxy ARP support for IPoE subscribers. When proxy ARP is enabled, if the BNG device receives an ARP request from Subscriber for which it has a route to the target (destination) IP address, the BNG device responds by sending a proxy ARP reply packet containing its own MAC address. The host/subscriber that sent the ARP request then sends the actual destined packets to the BNG, which forwards them to the intended destination. **Default**: NONE.
static-ipv4	The attribute static-ipv4 assigns a fixed static IPv4 address to all clients using this profile. ⚠ This feature should be only used with caution.
dad-enable	Enable/disable IPv4 duplicate address detection **Default:** true

Configuring IPv6

The address family IPv6 must be explicitly enabled in the access profile to be available for access protocols like PPP (PPPoE) or DHCP (IPoE).

```
supervisor@switch: cfg> set access access-profile pppoe-dual address-family ipv6
  <cr>
  enable                       Enable IPv6
  pool-name                    Local IPv6 pool name
  prefix-delegation-pool-name  Local IPv6 prefix delegation pool name
  primary-dns                  Primary DNS server
  secondary-dns                Secondary DNS server
  dad-enable                   Enable/disable IPv6 duplicate address detection
(Enabled by default)
```

Attribute	Description
enable	Enable IPv6 **Default:** false
pool-name prefix-delegation-pool-name	The optional pool-name attribute allows to assign of the IPv6 prefix from a locally managed pool as described in Address Pool Configuration. This prefix is advertised by ICMPv6 router-advertisements to the client where prefixes from optional prefix-delegation-pool-name are advertised by DHCPv6 as delegated prefix (IA_PD)
primary-dns secondary-dns	The primary-dns and secondary-dns servers configured are used by protocols like ICMPv6 router-advertisements or DHCPv6 and advertise to the client.
dad-enable	Enable/disable IPv6 duplicate address detection **Default:** true

IPv6 Router-Advertisement

```
supervisor@switch: cfg> set access access-profile pppoe-dual protocol ra
  <cr>
  enable                     Enable IPv6 router-advertisement
  interval                   Interval
  lifetime                   Lifetime
  preferred-lifetime         Preferred lifetime
```

Attribute	Description
enable	Enable IPv6 router-advertisement. **Default:** false
interval	IPv6 router-advertisements interval in seconds. **Default:** 0 (disabled)
lifetime	The valid lifetime for the prefix in seconds. **Default:** 14400
preferred-lifetime	The preferred lifetime for the prefix in seconds. **Default:** 1800

DHCPv4

```
supervisor@switch: cfg> set access access-profile ipoe-dual protocol dhcp
  <cr>
  enable                    Enable DHCP
  lease-time                DHCP lease time in seconds
  mode                      DHCP mode
```

Attribute	Description
enable	Enable DHCP. **Default:** false
dhcp-mode	This option defines the DHCP mode where the server handles DHCP requests locally and relay/proxy forwards those to the configured servers. The only difference between relay and proxy is the second one will hide the actual DHCP server. **Default:** server **Values:** server, relay, proxy ℹ Proxy mode is not supported now.
lease-time	The lease time for the address in seconds. **Default:** 300
dhcp-server	Configure global DHCP server.

DHCPv6

```
supervisor@switch: cfg> set access access-profile pppoe-dual protocol dhcpv6
  <cr>
  enable                    Enable DHCPv6
  lifetime                  Lifetime
  preferred-lifetime        Preferred lifetime
  mode                      DHCPv6 mode
```

Attribute	Description
enable	Enable DHCPv6. **Default:** false

Attribute	Description
mode	This option defines the DHCPv6 mode where server handles DHCPv6 requests locally and relay/proxy forwards those to the configured servers. The only difference between relay and proxy is that second one will hide the actual DHCPv6 server. **Default:** server **Values:** server, relay, proxy
lifetime	The valid lifetime for IPv6 prefixes in seconds. **Default:** 14400
preferred-lifetime	The preferred lifetime for IPv6 prefixes in seconds. This value should be less or equal to the valid lifetime, otherwise, RBFS will adjust the preferred lifetime to be equal to the valid lifetime. The values for T1 and T2 are 0.5 and 0.8 times the shortest preferred-lifetime. **Default:** 1800
dhcpv6-server	Configure DHCPv6 server.

Configuring PPPoE and PPP

The protocol PPPoE must be explicitly enabled in the access profile in order to allow PPPoE sessions.

```
supervisor@switch: cfg> set access access-profile pppoe-dual protocol pppoe enable
true
```

PPPoE

The PPPoE configuration allows changing the default behavior of the PPPoE protocol.

```
supervisor@switch: cfg> set access access-profile pppoe-dual protocol pppoe
  <cr>
  delete-terminated     Delete terminated sessions immediately without waiting for
subscriber daemon
  enable                Enable PPPoE
  max-outstanding       Maximum outstanding PPPoE sessions
  session-protection    PPPoE session protection
  vlan-priority         Control traffic VLAN priority code point (PCP)
```

Attribute	Description
enable	Enable PPPoE. **Default:** false
vlan-priority	Control traffic VLAN priority code point (PCP). **Default:** 0
delete-terminated	Delete terminated sessions immediately without waiting for the subscriber daemon. **Default:** false
max-outstanding	Maximum outstanding PPPoE sessions. **Default:** 64 **Range:** 1 - 65535

If PPPoE session protection is enabled, short-lived or failed sessions will be logged. Every session not established for at least 60 seconds per default (min-uptime) is considered a failed or short-lived session. This will block new sessions on this IFP and VLANs for one second per default (min-lockout), increasing exponentially with any further failed session until the maximum time of 300 seconds (max-lockout) is reached. The interval is reset after 900 seconds without failed sessions (currently not configurable).

PPPoE session protection logs the last subscriber-id and terminates the code which indicates the reason for session failures.

```
supervisor@switch: cfg> set access access-profile pppoe-dual protocol pppoe
session-protection
  <cr>
  enable                Enable PPPoE session protection
  max-lockout           Session protection maximum lockout time in seconds
  min-lockout           Session protection minimum lockout time in seconds
  min-uptime            Session protection minimum uptime in seconds
```

Attribute	Description
enable	Enable PPPoE session protection. **Default:** false

Attribute	Description
min-lockout	Session protection min lockout time (seconds). **Default:** 1
max-lockout	Session protection max lockout time (seconds). **Default:** 300
min-uptime	Session with an uptime less than this will trigger protection (seconds). **Default:** 60

PPP LCP

The PPP Link Control Protocol (LCP) configuration allows changing the default the behavior of the LCP protocol.

```
supervisor@switch: cfg> set access access-profile pppoe-dual protocol ppp lcp
  <cr>
  authentication-protocol  Authentication protocol
  config-nak-max           Max configure-reject/nak
  echo-enable              Enable echo requests
  echo-interval            Echo interval in seconds
  echo-max-retransmit      Echo maximum retries
  lcp-loop-detection       Loop detection
  mru                      Advertised local MRU
  mru-negotiation          MRU negotiation
  mtu                      Enforce this MTU as peer MRU
  retransmit-interval      Retransmit interval in seconds
  retransmit-max           Maximum retries
```

Attribute	Description
authentication-protocol	Per default, PPP authentication is set to NONE, which means disabled. This can be changed by setting the authentication protocol to either PAP or CHAP. The Password Authentication Protocol (PAP) is defined in RFC 1334 and receives the password as a plaintext value from the client. The Challenge Handshake Authentication Protocol (CHAP) is defined in RFC 1994 provides a more secure way to authenticate the client without exchanging plaintext secrets. The option PAP_CHAP offers the first PAP with a fallback to CHAP if PAP is rejected by the client. Alternative the option CHAP_PAP, which starts with CHAP falling back to PAP if CHAP is rejected by the client. **Default:** PAP_CHAP
echo-enable	Per default, RBFS will respond to LCP echo requests received but does not send until echo-enable is set to true. **Default:** false
echo-interval	LCP echo request interval in seconds. **Default:** 30 **Range:** 1 - 255
echo-max-retransmit	LCP echo request retransmissions. **Default:** 3 **Range:** 1 - 255
mru-negotiation	Negotiate MRU **Default:** true
mru	Local MRU (peer MTU) **Default:** 1492 **Range:** 256 - 1492
mtu	Local MTU (peer MRU) If set, this MTU is enforced as peer MRU, meaning that other values received will be rejected, proposing this value. **Default:** accept all **Range:** 256 - 1492

Attribute	Description
lcp-loop-detection	The negotiation and validation of magic numbers are enabled per default and can be disabled by setting lcp-loop-detection to false. It is not recommended to change this option! **Default:** true
retransmit-interval	The LCP request retransmit interval. **Default:** 5 **Range:** 1 - 255
retransmit-max	The LCP request retransmission before the session is terminated if no response is received. **Default:** 3 **Range:** 1 - 255
config-nak-max	The option config-nak-max defines the maximum PPP LCP configuration reject/nak messages that can be sent or received before the session is terminated. **Default:** 16 **Range:** 1 - 255

PPP IPCP

The address-family ipv4 and the protocol ppp ipcp must be explicitly enabled to use IPv4 over PPPoE. Additionally, the mandatory source-ifl option must be configured to derive the local IPv4 address from this logical interface.

```
supervisor@switch: cfg> set access access-profile pppoe-dual protocol ppp ipcp
  <cr>
  config-nak-max        Max configure-reject/nak
  enable                Enable PPP IPCP
  passive               Passive mode
  retransmit-interval   Retransmit interval in seconds
  retransmit-max        Maximum retries
  source-ifl            Source IFL
```

Attribute	Description
enable	Enable IPCP **Default:** false

Attribute	Description
passive	IPCP passive mode **Default:** false
source-ifl	This mandatory option must be configured to derive the local IPv4 address from this logical interface. This option should be set to the loopback interface of the corresponding routing instance.
retransmit-interval	The IPCP request retransmit interval. **Default:** 5 **Range:** 1 - 255
retransmit-max	The IPCP requests retransmission before the session is terminated if no response is received. **Default:** 8 **Range:** 1 - 255
config-nak-max	The option config-nak-max defines the maximum PPP IPCP configuration reject/nak messages that can be sent or received before the session is terminated. **Default:** 8 **Range:** 1 - 255

PPP IP6CP

Both the address-family ipv6 and the protocol ppp ip6cp must be explicitly enabled in order to use IPv6 over PPPoE.

```
supervisor@switch: cfg> set access access-profile pppoe-dual protocol ppp ip6cp
  <cr>
  config-nak-max        Max configure-reject/nak
  enable                Enable PPP IP6CP
  passive               Passive mode
  retransmit-interval   Retransmit interval in seconds
  retransmit-max        Maximum retries
```

Attribute	Description
enable	Enable IP6CP **Default:** false

Attribute	Description
passive	IP6CP passive mode **Default:** false
retransmit-interval	The IP6CP request retransmit interval. **Default:** 5 **Range:** 1 - 255
retransmit-max	The IP6CP requests retransmission before the session is terminated if no response is received. **Default:** 8 **Range:** 1 - 255
config-nak-max	The option config-nak-max defines the maximum PPP IP6CP configuration reject/nak messages that can be sent or received before the session is terminated. **Default:** 6 **Range:** 1 - 255

DHCP Server Configuration

You can configure the DHCP server in a routing instance by using the following commands. The following sections provide information about both DHCPv4 server and DHCPv6 server configurations.

Configuring DHCPv4 Server

```
supervisor@switch: cfg> set access dhcp-server
  dhcp-server            Global DHCP server configuration

supervisor@switch: cfg> set access dhcp-server server-example
  <cr>
  address                DHCP server address
  routing-instance       Instance name from which DHCP server is reachable
  source-address         Source address used for DHCP packets
```

The following example shows a DHCPv4 configuration.

```
supervisor@switch: cfg> show config access dhcp-server
{
  "rtbrick-config:dhcp-server": [
    {
      "server-name": "dhcp-server1",
      "address": "172.16.0.10",
      "source-address": "172.16.0.2",
      "routing-instance": "default"
    }
  ]
}
```

Configuring DHCPv6 Server

```
supervisor@switch: cfg> set access dhcpv6-server dhcpv6-example
  <cr>
  address                 DHCP server address
  routing-instance        Instance name from which DHCP server is reachable
  source-address          Source address used for DHCP packets
```

The following example shows a DHCPv6 configuration.

```
supervisor@switch: cfg> show config access dhcpv6-server
{
  "rtbrick-config:dhcpv6-server": [
    {
      "server-name": "dhcpv6-server1",
      "address": "172:16::2",
      "source-address": "172:16::10",
      "routing-instance": "default"
    }
  ]
}
```

AAA Profile Configuration

Table: global.access.aaa.profile.config

Subscriber management requires the mandatory configuration of an Authentication, Authorization, and Accounting (AAA) profile.

The way that the AAA profile configuration relates to all subscriber management configuration tasks are shown in the picture below.

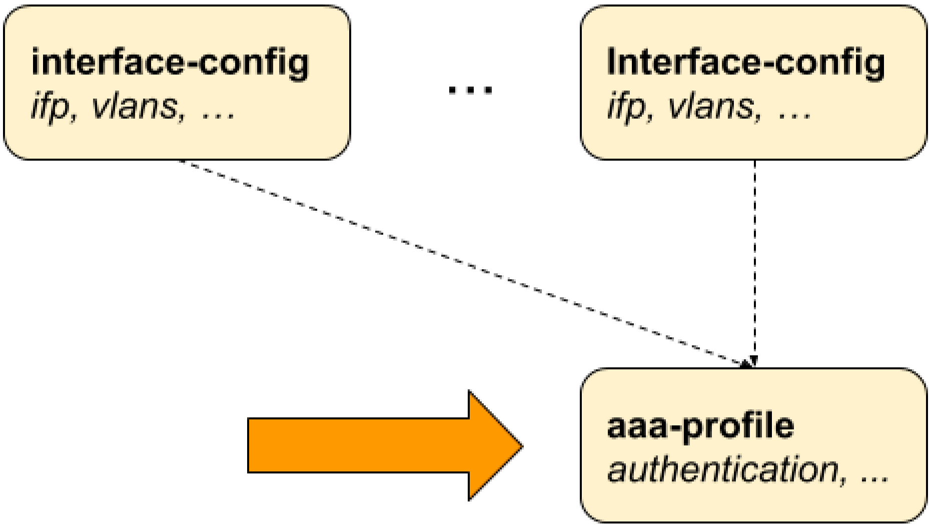


Figure 10. AAA Profile Configuration

Configuring the AAA Profile

```
supervisor@switch: cfg> set access aaa-profile
  <profile-name>        Name of the AAA profile

supervisor@switch: cfg> set access aaa-profile aaa-example
  <cr>
  aaa-radius-profile    AAA RADIUS profile name
  accounting            Accounting options
  authentication        Authentication options
  idle-timeout          Idle timeout in seconds (0 == infinity)
  session-timeout       Session timeout in seconds (0 == infinity)
```

The following example shows a typical AAA profile for RADIUS authentication and accounting.

```
supervisor@switch: cfg> show config access aaa-profile aaa-radius
{
  "rtbrick-config:aaa-profile": {
    "profile-name": "aaa-radius",
    "session-timeout": 0,
    "idle-timeout": 0,
    "aaa-radius-profile": "radius-default",
    "authentication": {
      "order": "RADIUS"
    },
    "accounting": {
      "order": "RADIUS",
      "session-id-format": "DEFAULT",
      "ingress": {
        "accounting-source": "POLICER"
      },
      "egress": {
        "accounting-source": "CLASS",
        "class-byte-adjustment-value": 16
      }
    }
  }
}
```

Attribute	Description
session-timeout	The session timeout specifies the maximum uptime in seconds until a subscriber is terminated. The value 0 means infinity. **Default:** 0 **Range:** 0 - 4294967295
idle-timeout	The idle timeout specifies the time in seconds until a subscriber is terminated if no traffic is forwarded, based on outgoing logical interface statistics of the subscriber IFL. Those statistics do not include control traffic. The subscriber is not considered idle as long as egress traffic is detected. The idle timeout is not limited but should be set to at least double the time of the logical interface statistics counter update interval (between 5 to 30 seconds). The value 0 means infinity. **Default:** 0 **Range:** 0 - 4294967295
aaa-radius-profile	The RADIUS profile (RADIUS Profile Configuration) which is used if RADIUS authentication or accounting is enabled.

Configuring Authentication

RBFS supports the authentication methods NONE, LOCAL, DOMAIN, and RADIUS. The option NONE disables authentication by accepting all credentials. The authentication method LOCAL authenticates the subscriber based on locally defined user profiles User Profile Configuration. The method DOMAIN works similarly to LOCAL, but except for the whole username, only the domain part separated by a configurable domain delimiter (default @)is used like rtbrick.com for user user@rtbrick.com. The authentication method RADIUS authenticates the subscriber remotely by sending an authentication request to the defined RADIUS servers.

The authentication method DOMAIN is currently not supported!

Some methods can also be combined together. With LOCAL_RADIUS the subscriber is first authenticated locally and secondly via RADIUS if no matching local user is found. The subscriber is immediately rejected without requesting RADIUS servers if a local user is found, but the password does not match. The behavior is similar for RADIUS_LOCAL where the subscriber is immediately disconnected if the authentication request is rejected by RADIUS. In this case, local authentication is used as the fallback if no response is received (timeout) from any RADIUS server configured.

```
supervisor@switch: cfg> set config access aaa-profile aaa-default authentication
  <cr>
  delimiter                Delimiter string
  order                    Authentication order
```

Attribute	Description
order	This option defines the order of authentication methods. **Default:** NONE **Values:** LOCAL, LOCAL_RADIUS, RADIUS, RADIUS_LOCAL
delimiter	This option defines the delimiter for domain authentication. **Default:** @ ℹ Currently not supported!

Configuring Accounting

Subscriber accounting refers to the process of measuring and recording the time and data usage of an corresponding subscriber. This includes the session time, called time accounting, and the number of packets and bytes transmitted or received called volume accounting.

Subscriber volume accounting works in both directions, but ingress and egress direction can be configured independently.

ℹ Today RBFS supports the accounting method RADIUS only!

```
supervisor@switch: cfg> set config access aaa-profile aaa-default accounting
  <cr>
  egress                   Egress volume accounting options
  ingress                  Ingress volume accounting options
  interim-interval         Accounting interim interval in seconds (0 == disabled)
  order                    Accounting order
  session-id-format        Accounting-Session-Id format
```

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td>order</td><td>This option defines the order of accounting methods.

<b>Default:</b> NONE</td></tr>
<tr><td>interim-interval</td><td>The interim interval specifies the time between interim accounting requests in seconds where 0 means disabled.

<b>Default:</b> 0 <b>Range:</b> 0 - 4294967295</td></tr>
<tr><td>session-id-format</td><td>The format of the Accounting-Session-Id (RADIUS attribute 44).

<table>
<tr><th>Name</th><th>Format</th><th>Example</th></tr>
<tr><td>DEFAULT</td><td>&lt;subscriber-id>.<timestamp&gt;</td><td>72339069014639577.1551943760</td></tr>
<tr><td>BRIEF</td><td>&lt;subscriber-id&gt;&gt;</td><td>72339069014639577</td></tr>
<tr><td>EXTENSIVE</td><td>&lt;subscriber-id>.<ifp>.<outer-vlan>.<inner-vlan>.<client-mac>.<session-id>.<timestamp&gt;</td><td>72339069014639577.ifp-0/0/0.128.7.01:02:03:04:05:05.1.1551943760</td></tr>
</table>

<b>Default:</b> DEFAULT <b>Values:</b> BRIEF, EXTENSIVE

ℹ Currently, only DEFAULT is supported!</td></tr>
</table>

Configuring Accounting Adjustments

The accounting adjustment feature enables basic counter modifications for the configured accounting method, such as RADIUS accounting. This configuration is necessary to normalize counters across different platforms in each direction. On Broadcom Q2C and Q2A based platforms, packets are counted in the size they enter the switch. Without adjustment, egress accounting would count downstream traffic as received from the core, complete with MPLS labels, while ingress accounting typically includes VLAN headers and/or PPPoE headers.

This counter adjustment aims to normalize counters with diverse encapsulations (double-tagged, untagged, etc.), potentially aligning to L3 counters (IP header and payload) as an example, or exclusively adapting egress traffic to match the outgoing packet encapsulation. The possibility for seperate adjustment configurations per direction allows parity in the counters for both ingress and egress.

Within RBFS, there are two configurations available for this purpose: the byte adjustment value and the factor, with the latter rarely needed. The byte adjustment value accommodates both positive and negative values, like -20.0 or 20.0. Any provided decimal digits in the adjustment values are ignored (e.g. 20.2 becomes 20.0). The byte adjustment factors accept positive values and utilize only the first two decimal places, such as 0.98 (-2%) or 1.02 (+2%).

Ingress Accounting

Subscriber ingress accounting refers to the process of measuring and recording the data usage or traffic that enters a subscriber interface (upstream).

```
supervisor@switch: cfg> set config access aaa-profile aaa-default accounting
ingress
  <cr>
  accounting-source               Source of session ingress counter
  byte-adjustment-factor          Adjust ingress LIF counters by factor
  byte-adjustment-value           Adjust ingress LIF counters by N bytes per
packet
  policer-byte-adjustment-factor  Adjust ingress policer counters by factor
  policer-byte-adjustment-value   Adjust ingress policer counters by N bytes per
packet
```

Attribute	Description
accounting-source	This option provides control over the counters used for subscriber ingress accounting when RADIUS accounting is enabled. The counters in question are the RADIUS attributes Acct-Input-Packets (47), Acct-Input-Octets (42), and Acct-Input-Gigawords (52). By default, the policer statistics (POLICER) are utilized, which represent the total traffic accepted across all policer levels (1-4). However, ingress control traffic is subject to a separate control plane policer and is therefore not included in the session policer statistics. Consequently, policers are necessary if session accounting is required. Alternatively, the logical interface (LIF) statistics can be employed, encompassing all received traffic, including control traffic and traffic dropped by the ingress policer. It is important to note that this option may not be available on all platforms. **Default:** POLICER **Values:** POLICER, LIF
byte-adjustment-value	Adjust ingress LIF counters by +/- N bytes per packet. **Default:** 0.00 **Range:** -32 - 32
byte-adjustment-factor	Adjust ingress LIF counters by a factor (executed after adjustment value). **Default:** 1.00 **Range:** 0.00 - 2.00
policer-byte-adjustment-value	Adjust ingress POLICER counters by +/- N bytes per packet. **Default:** 0.00 **Range:** -32 - 32
policer-byte-adjustment-factor	Adjust ingress POLICER counters by factor (executed after adjustment value). **Default:** 1.00 **Range:** 0.00 - 2.00

Egress Accounting

Subscriber egress accounting refers to the process of measuring and recording the data usage or traffic that is sent from a subscriber interface (downstram).

```
supervisor@switch: cfg> set config access aaa-profile aaa-default accounting
egress
  <cr>
  accounting-source              Source of session egress counter
  byte-adjustment-factor         Adjust egress LIF counters by a factor
  byte-adjustment-value          Adjust egress LIF counters by N bytes per packet
  class-byte-adjustment-factor   Adjust egress class counters by a factor
  class-byte-adjustment-value    Adjust egress class counters by N bytes per packet
```

Attribute	Description
accounting-source	This option provides control over the counters used for egress session accounting when RADIUS accounting is enabled. The counters in question are the RADIUS attributes Acct-Output-Packets (48), Acct-Output-Octets (43), and Acct-Output-Gigawords (53). By default, the class statistics (CLASS) are utilized, which represent the total traffic accepted across all queues. However, the egress control traffic is sent directly to the IFP and is therefore not included in the session class statistics. Consequently, QoS is necessary if session accounting is required. As an alternative, the logical interface (LIF) statistics can be utilized, which cover all sent traffic, excluding control traffic. However, it is important to be aware that this option might not be accessible on all platforms. **Default:** CLASS **Values:** CLASS, LIF
byte-adjustment-value	Adjust egress LIF counters by +/- N bytes per packet. **Default:** 0.00 **Range:** -32 - 32
byte-adjustment-factor	Adjust egress LIF counters by a factor (executed after adjustment value). **Default:** 1.00 **Range:** 0.00 - 2.00

Attribute	Description
class-byte-adjustment-value	Adjust egress CLASS (queue) counters by +/- N bytes per packet. **Default:** 0.00 **Range:** -32 - 32
class-byte-adjustment-factor	Adjust egress CLASS (queue) counters by factor (executed after adjustment value). **Default:** 1.00 **Range:** 0.00 - 2.00

RADIUS Profile Configuration

Subscriber management allows the configuration of a RADIUS profile which is mandatory if RADIUS is used for authentication or accounting.

The way that the RADIUS profile configuration relates to all subscriber management configuration tasks is shown in the picture below.

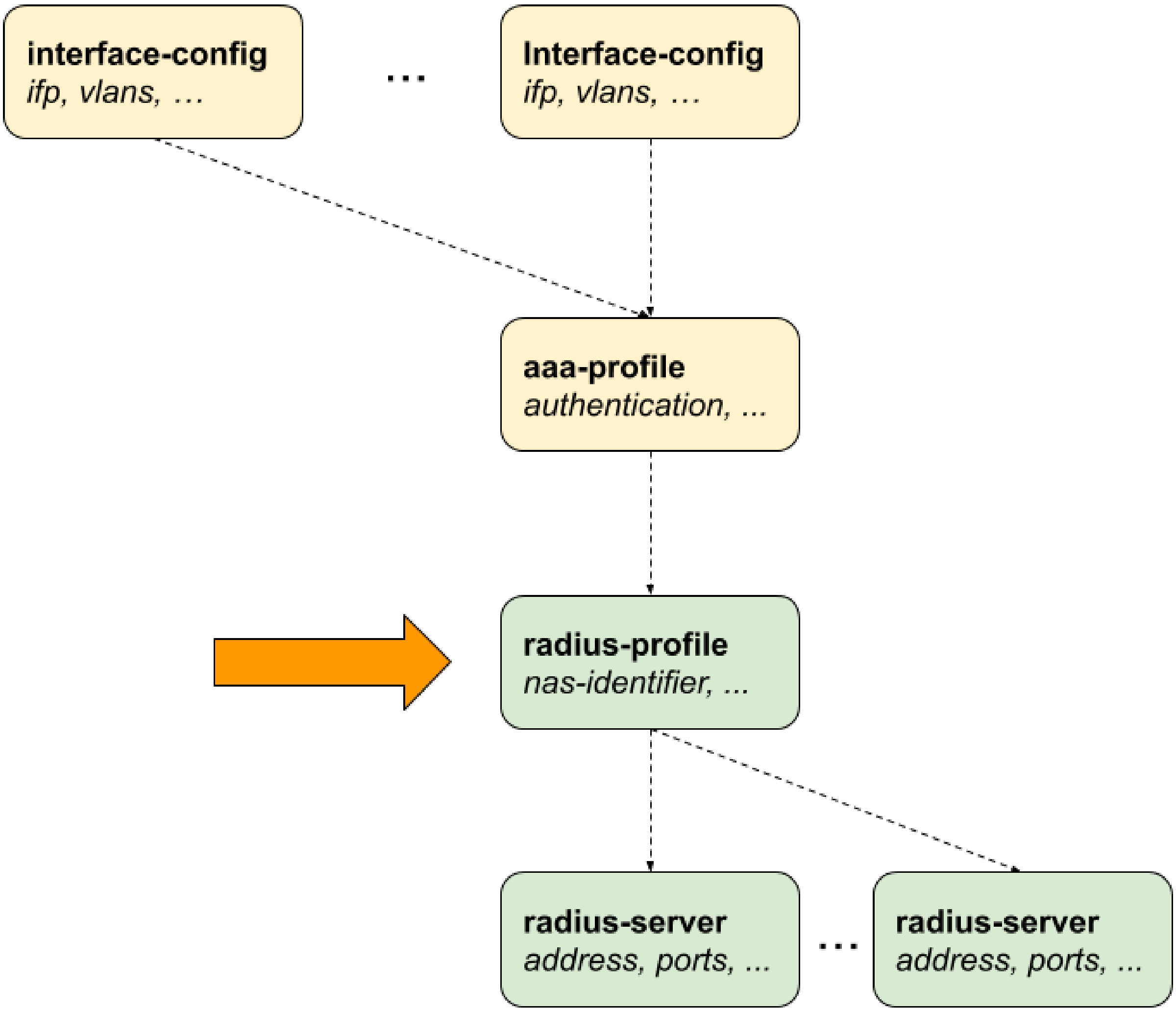


Figure 11. RADIUS Profile Configuration

Configuring the RADIUS Profile

```
supervisor@switch: cfg> set config access radius-profile
  <profile-name>        Name of the RADIUS profile

supervisor@switch: cfg> set config access radius-profile radius-default
  <cr>
  accounting            RADIUS accounting options
  authentication        RADIUS authentication options
  nas-identifier        NAS identifier
  nas-ip-address        NAS IP address (IPv4 Address)
  nas-port-format       NAS-Port format
  nas-port-type         NAS-Port type
```

The following example shows a typical RADIUS profile for authentication and accounting.

```
supervisor@switch: cfg> show config access radius-profile radius-default
{
  "rtbrick-config:radius-profile": {
    "profile-name": "radius-default",
    "nas-identifier": "BNG",
    "nas-port-type": "Ethernet",
    "authentication": {
      "radius-server-profile-name": [
        "radius-server-1",
        "radius-server-2"
        ]
    },
    "accounting": {
      "radius-server-profile-name": [
        "radius-server-1",
        "radius-server-2"
        ],
      "stop-on-reject": "true",
      "stop-on-failure": "true",
      "accounting-on-off": "true",
      "accounting-on-wait": "true",
      "accounting-backup": "true",
      "accounting-backup-max": 86400
    }
  }
}
```

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td>nas-identifier</td><td>Set the value for the RADIUS attribute NAS-Identifier (32).

<b>Default:</b> system hostname</td></tr>
<tr><td>nas-ip-address</td><td>Set the value for RADIUS attribute NAS-IP-Address (4).

<b>Default</b>: source IPv4 address</td></tr>
<tr><td>nas-port-type</td><td>Set the value for RADIUS attribute NAS-Port-Type (61).

<b>Default:</b> Ethernet</td></tr>
<tr><td>nas-port-format</td><td>Set the format of the 32-bit RADIUS attribute NAS-Port (5).
<table>
<tr><th>Name</th><th>Bits</th><th>Values</th></tr>
<tr><td>DEFAULT</td><td>1:1:6:12:12</td><td>slot:subslot:port:vlan:vlan</td></tr>
<tr><td>SLOTS</td><td>6:2:6:12:6</td><td>slot:subslot:port:vlan:vlan</td></tr>
</table>
</td></tr>
</table>

Configuring Authentication

```
supervisor@switch: cfg> set config access radius-profile radius-default
authentication
  <cr>
  algorithm-type              Authentication redundancy algorithm
  radius-server-profile-name  RADIUS server profile name
```

Attribute	Description
radius-server-profile-name	List of RADIUS servers used for authentication.
algorithm-type	Authentication server selection algorithm as described in RADIUS Redundancy. **Default:** DIRECT **Values:** DIRECT, ROUND-ROBIN

Configuring Accounting

```
supervisor@switch: cfg> set config access radius-profile radius-default accounting
  <cr>
  accounting-backup            Enable backup accounting
  accounting-backup-max        Max backup accounting hold time in seconds
  accounting-on-off            Enable accounting on/off
  accounting-on-wait           Wait for an accounting-on response before sending
authentication requests
  algorithm-type               Accounting redundancy algorithm
  radius-server-profile-name   RADIUS server profile name
  stop-on-failure              Send accounting-stop on failure
  stop-on-reject               Send accounting-stop on authentication reject
```

Attribute	Description
radius-server-profile-name	List of RADIUS servers used for accounting.
algorithm-type	Accounting server selection algorithm as described in RADIUS Redundancy. **Default:** DIRECT **Values:** DIRECT, ROUND-ROBIN
stop-on-failure	Sent RADIUS accounting request stop in case of failure after authentication was accepted. **Default:** false
stop-on-reject	Sent RADIUS accounting request stop in case of authentication is rejected. **Default:** false

Attribute	Description
accounting-on-off	Enable RADIUS Accounting-On/Off messages as described in RADIUS Accounting. **Default:** false
accounting-on-wait	This option prevents any new subscriber until the accounting hast started meaning that the Accounting-On response was received. **Default:** false
accounting-backup	RADIUS accounting requests are often used for billing and, therefore should be able to store and retry over a longer period (common up to 24 hours or more) which can be optionally enabled here. **Default:** false
accounting-backup-max	This option defines maximum backup accounting hold time in seconds if accounting backup is enabled. **Default:** 3600 **Range:** 1 - 4294967295

RADIUS Server Configuration

Successful subscriber management AAA methods are often supplied by a RADIUS server, although there are cases where other forms of AAA, including local methods independent of network availability, are appropriate.

RADIUS server configuration is a *dependent* step in subscriber management configuration. In other words, if you configure an optional RADIUS profile for AAA, then you must configure a RADIUS server to go along with it. So, RADIUS server configuration is dependent on RADIUS profile configuration.

The way that the RADIUS server configuration relates to all subscriber management configuration tasks is shown in the picture below.

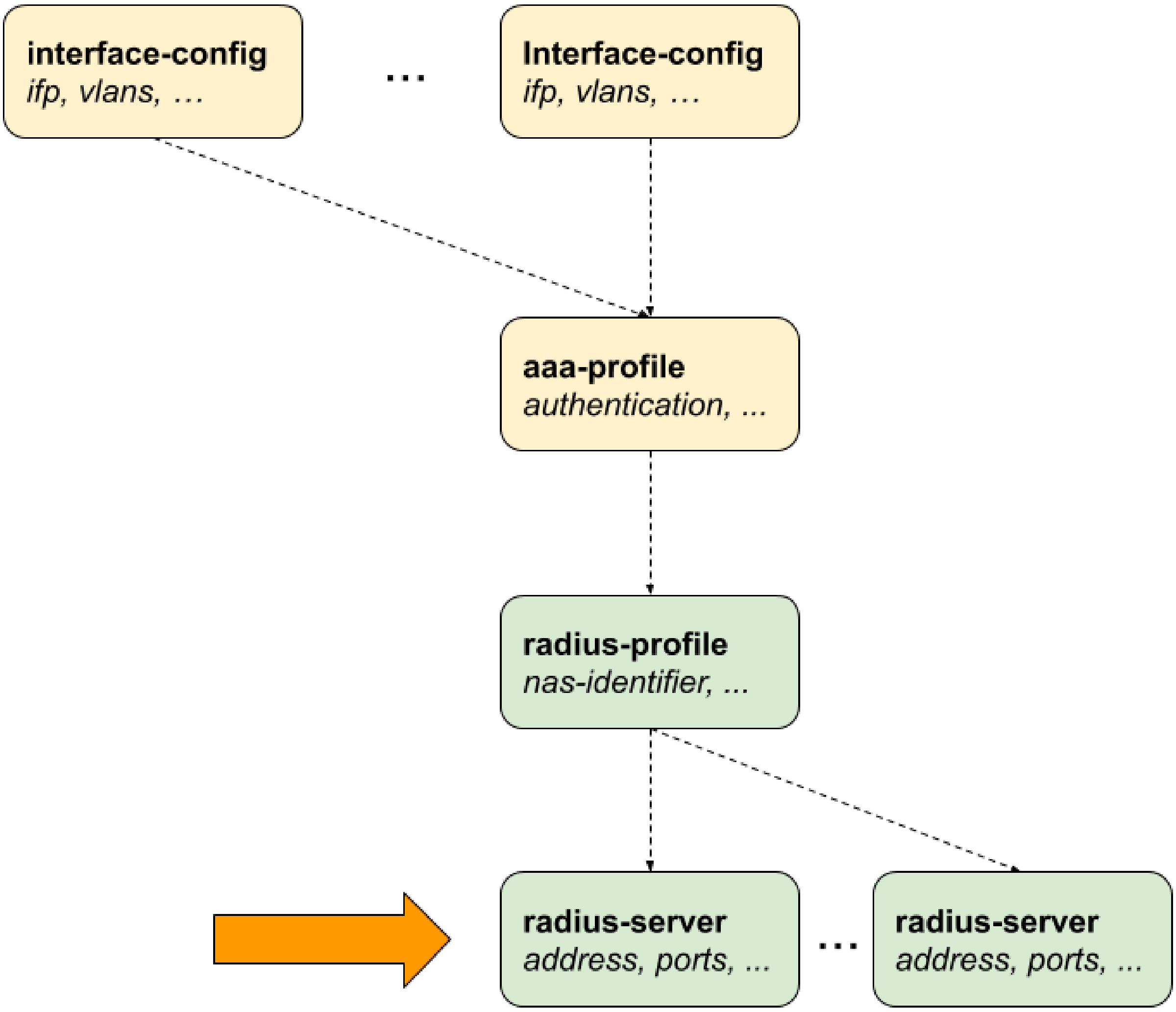


Figure 12. RADIUS Server Configuration

Configuring the RADIUS Server

```
supervisor@switch: cfg> set config access radius-server
  <server-name>          Name of the RADIUS server

supervisor@switch: cfg> set config access radius-server radius-server-1
  <cr>
  accounting              RADIUS accounting mode
  address                 RADIUS server address
  authentication          RADIUS authentication mode
  coa                     RADIUS Change-of-Authorization (CoA) mode
  rate                    Maximum RADIUS requests per/second
  routing-instance        Instance name
  secret-encrypted-text   RADIUS secret in encrypted text
  secret-plain-text       RADIUS secret in plain text
  source-address          Source address used for RADIUS packets
```

The following example shows a typical ...

```
supervisor@switch: cfg> show config access radius-server radius-server-1
```

```
{
  "rtbrick-config:radius-server": {
    "server-name": "radius-server-1",
    "address": "198.51.100.101",
    "source-address": "198.51.100.200",
    "secret-encrypted-text": "$21e4946e31b406de98b3077aef03ed5a7",
    "authentication": {
      "enable": "true"
    },
    "accounting": {
      "enable": "true"
    },
    "coa": {
      "enable": "true"
    }
  }
}
```

Attribute	Description
address	RADIUS server IPv4 address. *Multiple RADIUS servers with the same IPv4 address are currently not supported, even if the instance or port is different.!*
source-address	Local source IPv4 address.
routing-instance	The routing instance in which the RADIUS server is reachable.
secret-encrypted-text secret-plain-text	RADIUS secret, which can be provided as plaintext or already encrypted text.
rate	Maximum RADIUS requests per second. **Default:** 600 **Range:** 1 - 65535

Configuring Authentication

```
supervisor@switch: cfg> set access radius-server radius-server-1 authentication
  <cr>
  enable                    Enable RADIUS authentication
  outstanding               Maximum number of outstanding authentication requests
  port                      RADIUS server authentication port
  retry                     Maximum retries for authentication request packets
  timeout                   Authentication request timeout in seconds
```

Attribute	Description
enable	Enable RADIUS authentication. **Default:** false
port	RADIUS authentication port. **Default:** 1812 **Range:** 1 - 65535
retry	This option specifies the number of authentication retries before declaring this server as unreachable for authentication. After reaching the limit, the client begins to send requests to other RADIUS servers and rejects the request after receiving the end of the list. **Default:** 3 **Range:** 1 - 255
timeout	Authentication request timeout in seconds. **Default:** 5 **Range:** 1 - 65535
outstanding	This option specifies the maximum number of outstanding authentication requests for this RADIUS server. A request is counted as outstanding if sent out but the response is not received. **Default:** 100 **Range:** 1 - 65535

Configuring Accounting

```
supervisor@switch: cfg> set access radius-server radius-server-1 accounting
  <cr>
  enable                Enable RADIUS accounting
  outstanding           Maximum number of outstanding accounting requests
  port                  RADIUS server accounting port
  retry                 Maximum retries for accounting request packets
  timeout               Accounting request timeout in seconds
```

Attribute	Description
enable	Enable RADIUS accounting. **Default:** false

Attribute	Description
port	RADIUS authentication port. **Default:** 1813 **Range:** 1 - 65535
retry	This option specifies the number of accounting retries before declaring this server as unreachable for accounting. After reaching the limit, the client begins to send requests to other RADIUS servers. **Default:** 10 **Range:** 1 - 255
timeout	Authentication request timeout in seconds. **Default:** 30 **Range:** 1 - 65535
outstanding	This option specifies the maximum number of outstanding accounting requests for this RADIUS server. A request is counted as outstanding if sent out, but the response is not received. **Default:** 100 **Range:** 1 - 65535

Configuring Change-of-Authorization (CoA)

```
supervisor@switch: cfg> set access radius-server radius-server-1 coa
  <cr>
  enable                Enable Change-of-Authorization (CoA)
  port                  Local RADIUS CoA port
```

Attribute	Description
enable	Enable receiving of RADIUS CoA requests from this server. **Default:** false
port	RADIUS CoA port. **Default:** 3799 **Range:** 1 - 65535

Service Profile Configuration

Service profile configuration is an optional step in subscriber management configuration which allows to assign QoS or IGMP configurations to a subscriber.

The way that the service profile configuration relates to all subscriber management configuration tasks is shown in the picture below.

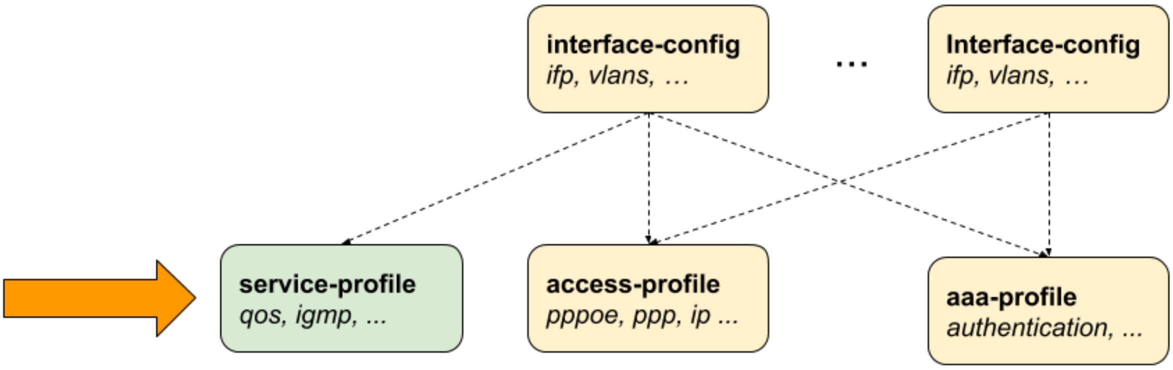


Figure 13. Service Profile Configuration

Configuring the Service Profile

```
supervisor@switch: cfg> set access service-profile
  <profile-name>         Name of the service profile

supervisor@switch: cfg> set access service-profile iptv
  <cr>
  igmp                   IGMP related attributes
  qos                    QoS related attributes
```

The following example shows a typical service profile for subscribers with IPTV (multicast) services.

```
supervisor@switch: cfg> show config access service-profile iptv
{
  "rtbrick-config:service-profile": {
    "profile-name": "iptv",
    "qos": {
      "profile": "iptv-qos-xl"
    },
    "igmp": {
      "enable": "true",
      "profile": "iptv-basic",
      "version": "IGMPv3",
      "max-members": 10
    }
  }
}
```

Configuring QoS

```
supervisor@switch: cfg> set access service-profile iptv qos
  <cr>
  parent-scheduler        QoS parent scheduler
  profile                 QoS profile
```

Attribute	Description
parent-scheduler	This options defines the parent scheduler element of the scheduler-map which is assigned to the subscriber. If not present, the scheduler-map will be directly bound to the local IFP where the session is established. This attribute can be only set once and never be changed without disconnect of the session. The parent scheduler can be also set via RADIUS which has priority over the one defined here. ⚠ Providing a QoS parent scheduler which is not present on the corresponding IFP will lead to blackholing of all egress data traffic. Control traffic is not impacted and therefore the session will remain.
profile	This option assigns a QoS configuration profile to the subscriber. The QoS profile can be also set via RADIUS which has priority over the one defined here.

Configuring IGMP

```
supervisor@switch: cfg> set access service-profile iptv igmp
  <cr>
  enable                  Enable IGMP service
  max-members             Maximum IGMP membership per subscriber
  profile                 IGMP profile
  version                 IGMP version
```

Attribute	Description
enable	This attribute dynamically enables or disables IGMP for a subscriber. **Default:** false

Attribute	Description
max-members	This attribute limits the number of parallel multicast channels (maximum IGMP membership) for a subscriber. **Default:** 1 **Range:** 1 - 4294967295
profile	This attribute specifies the IGMP profile to be associated with the subscriber.
version	This attribute can specify the version of IGMP for a subscriber. **Default:** V3 **Values:** V1, V2, V3

L2TP Profile Configuration

The Layer 2 Tunnel Protocol (L2TPv2) profile configuration is an optional step in subscriber management configuration which is mandatory to enable L2TP tunneling.

The picture below illustrates how all subscriber management configuration tasks are related to L2TP profile configuration.

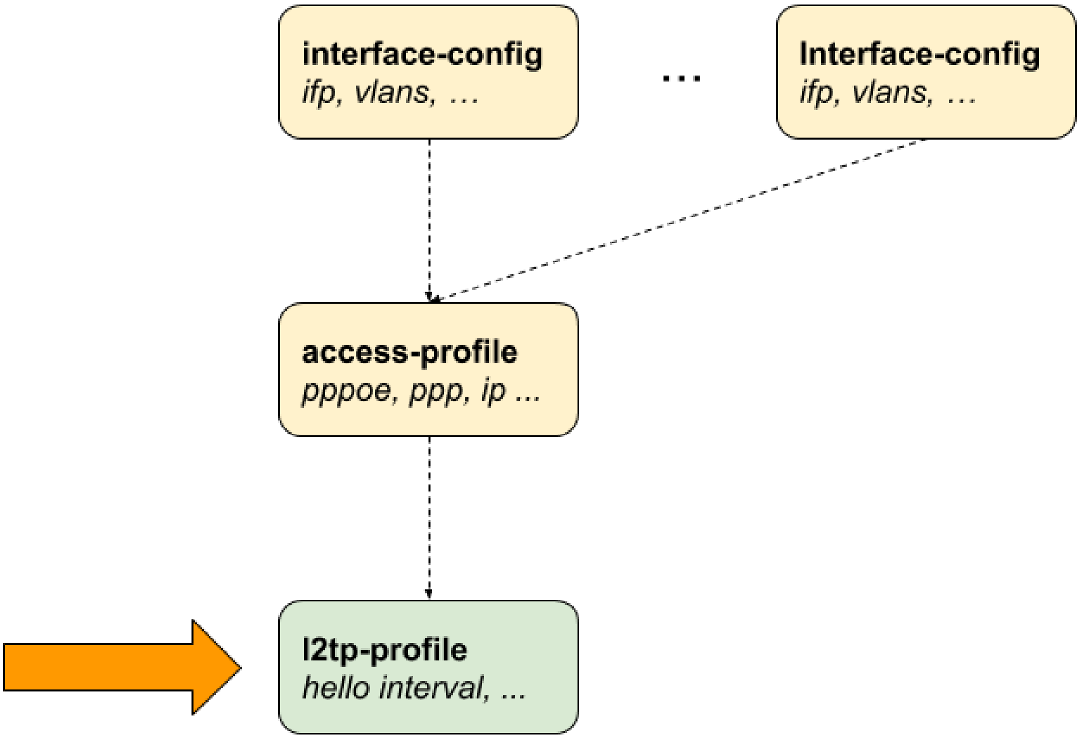


Figure 14. L2TPv2 Profile Configuration

Configuring the L2TP Profile

```
supervisor@switch: cfg> set access l2tp-profile
  <profile-name>        Name of the L2TP profile

supervisor@switch: cfg> set access l2tp-profile l2tp-default
  <cr>
  client-ipv4                Default value for L2TP tunnel client IPv4 address
  client-name                Default value for L2TP tunnel client name
  connect-speed-update       Enable L2TP Connect-Speed-Update-Notification (CSUN)
  dead-timeout-interval      L2TP tunnel dead timeout interval in seconds
  hello-interval             L2TP tunnel hello interval in seconds
  hide-authentication        Hide L2TP tunnel authentication
  idle-timeout-interval      L2TP tunnel idle timeout interval in seconds
  inactive-timeout-interval  L2TP tunnel inactive timeout interval in seconds
  instance                   Instance name
  pon-access-line-version    PON Access Line Information Version
  pool-name                  L2TP tunnel pool name
  receive-window             L2TP tunnel receive window
  request-retries            L2TP session request retries
  request-timeout-interval   L2TP session request timeout interval in seconds
  retransmit-interval        L2TP tunnel retransmission interval in seconds
  selection-algorithm        L2TP tunnel selection algorithm
  service-label              MPLS service label
  session-limit              L2TP tunnel session limit
```

The following example shows a typical L2TPv2 LAC configuration profile.

```
supervisor@switch: cfg> show config access l2tp-profile l2tp-default
{
  "rtbrick-config:l2tp-profile": {
    "profile-name": "l2tp-default",
    "session-limit": 4000,
    "hello-interval": 60,
    "client-name": "BNG",
    "client-ipv4": "198.51.100.200",
    "hide-authentication": true
    "service-label": 1234
  }
}
```

Attribute	Description
client-ipv4	This is the default value for the local L2TP tunnel client (LAC) IPv4 address if not explicitly provided for the tunnel via L2TP pool or RADIUS.
client-name	This is the default value for the local L2TP tunnel client (LAC) hostname if not explicitly provided for the tunnel via L2TP pool or RADIUS. **Default:** system hostname

Attribute	Description
instance	The routing instance in which the L2TP endpoint (LNS) is reachable. **Default:** default
service-label	The service label must be defined to support L2TP over MPLS Configuring L2TP over MPLS. Supported MPLS label values are 0 - 1048575. The reserved MPLS label range is 0 - 15. In RBFS, BGP uses the label range 20000 - 100000. It is recommended to assign label values outside of these reserved ranges to avoid conflicts.
selection-algorithm	This defines how to select a tunnel from a pool of available LNS servers as described in L2TP Tunnel Selection. The RANDOM algorithm selects the tunnel randomly, whereas BALANCED selects the least filled tunnel based on the number of sessions. **Default:**: BALANCED **Values:** BALANCED, RANDOM
session-limit	This is the default tunnel session limit if not further specified. Tunnels with session limit reached are not considered for further sessions. **Default:** 64000 **Range:** 1 - 65535
pool-name	This attribute allows to assign a default L2TP tunnel pool L2TP Tunnel Pool Configuration which can be overwritten by user-defined pool names from local user profiles User Profile Configuration or received via RADIUS attribute RtBrick-L2TP-Pool (VSA 26-50058-40).
hello-interval	L2TP tunnel hello interval in seconds where 0 means disabled. The HELLO keep alive messages are part of the L2TP control channel L2TP Control Channel and only send if there is no other message sent if the queue is empty and no other message is sent during the hello interval. **Default:** 30 **Range:** 0 - 86400

Attribute	Description
idle-timeout-interval	This interval defines the maximum time in seconds to keep a tunnel without sessions established. The session will remain forever if this value is set to 0. **Default:** 600 **Range:** 0 - 4294966
dead-timeout-interval	This interval defines the time in seconds to keep an unreachable tunnel in a DEAD state. After interval expiration the tunnel changes back to DOWN state to be available for new sessions. **Default:** 300 **Range:** 1 - 4294966
inactive-timeout-interval	This interval defines the time in seconds to keep an inactive tunnel before removal. This interval is reset with every new session request which considers this tunnel as a potential candidate. **Default:** 900 **Range:** 1 - 4294966
receive-window	This value specifies the receive window size being offered to the remote peer trough Receive Window Size AVP (10) in SCCRQ, SCCRP. Suppose advertising a receive window size of 8 in the SCCRQ or SCCRP messages. The remote peer is now allowed to have up to 8 outstanding control messages. Once eight have been sent; it must wait for an acknowledgment that advances the window before sending new control messages. **Default:** 8 **Range**: 1 - 256
request-retries	This value is explained together with request-timeout-interval. **Default:** 5 **Range**: 1 - 600

Attribute	Description
request-timeout-interval	This interval multiplied with the request-retries defines the maximum time in seconds to wait for the selected tunnel to become established before selecting another tunnel from the list. **Default:** 1 **Range:** 1 - 30 ⚠ The values for request-retries and request-timeout-interval should be changed with caution!
retransmit-interval	This value specifies the retransmission interval in seconds. Each subsequent retransmission of a message employ an exponential backoff interval. Thus, if the first retransmission occurred after 1 second; the next retransmission occurred after 2 seconds had elapsed, then 4 seconds, 8 seconds, 16 seconds, 32 seconds, and finally 64 seconds. This maximum value is reached after a maximum of 6 retransmissions resulting in max 64 seconds for a retransmit interval of 1, 128 seconds for 2, etc. **Default:** 1 **Range**: 1 - 30
hide-authentication	If enabled, the L2TP proxy authentication response AVP will be hidden if authentication type is PAP to not transmit the password in clear text. **Default:** false

Attribute	Description
pon-access-line-version	Adding additional PON attributes to the L2TP access line information L2TP Access Line Information (RFC5515) as defined in draft-lihawi-ancp-protocol-access-extension which can be optionally enabled using this configuration attribute. RFC and draft compliance are partial except as specified. The value DRAFT-LIHAWI-00 enables PON attributes based on the definition in draft-lihawi-ancp-protocol-access-extension-00 whereas DRAFT-LIHAWI-04 uses draft-lihawi-ancp-protocol-access-extension-04. **Default:**: DISABLED **Values:** DRAFT-LIHAWI-00, DRAFT-LIHAWI-04
connect-speed-update	

Configuring L2TP over MPLS

L2TP over MPLS requires a dedicated L2TP service label which needs to be configured manually.

Following an example L2TP configuration with L2TP service label.

```
set access l2tp-profile l2tp-default service-label 1234
```

Advertising this label via BGP must be configured manually as shown in the example below. The exact policy configuration depends on the actual network and existing policy concept.

```
supervisor@switch: cfg> show config policy
{
    "rtbrick-config:policy": {
      "statement": [
        {
          "name": "L2TP_MPLS",
          "ordinal": [
            {
              "ordinal": 1,
              "match": {
                "rule": [
                  {
                    "rule": 1,
                    "type": "ipv4-prefix",
                    "value-type": "discrete",
                    "match-type": "exact",
                    "value": "198.51.100.200/24"
                  }
                ]
              },
              "action": {
                "rule": [
                  {
                    "rule": 1,
                    "type": "label",
                    "operation": "overwrite",
                    "value": "label:1337,bos:1"
                  }
                ]
              }
            },
            {
              "ordinal": 2,
              "action": {
                "rule": [
                  {
                    "rule": 1,
                    "operation": "return-permit"
                  }
                ]
              }
            }
          ]
        }
      ]
    }
  }


supervisor@switch: cfg> show config instance internet
{
  "rtbrick-config:instance": {
    "name": "internet",
    "address-family": [
      {
        "afi": "ipv4",
        "safi": "unicast",
        "policy": {
          "export": "L2TP_MPLS"
        }
      }
    ]
  }
}
```

L2TP Tunnel Pool Configuration

The Layer 2 Tunnel Protocol (L2TPv2) pool configuration is an optional step in subscriber management configuration which allows to define local sets of possible

L2TP LNS server endpoints.

Configuring the L2TP Tunnel Pool

```
supervisor@switch: cfg> set access l2tp-pool
  <pool-name>           Name of the L2TP pool

supervisor@switch: cfg> set access l2tp-pool lns-servers
  <client-name>         L2TP client (LAC) name

supervisor@switch: cfg> set access l2tp-pool lns-servers BNG
  <server-name>         L2TP server (LNS) name

supervisor@switch: cfg> set access l2tp-pool lns-servers BNG LNS
  <cr>
  client-ipv4            L2TP client (LAC) IPv4
  preference             Preference
  secret-encrypted-text  Shared secret in encrypted text
  secret-plain-text      Shared secret in plain text
  server-ipv4            L2PTP server (LNS) IPv4
  session-limit          Session limit
```

The following example shows a local pool with two LNS severs.

```
supervisor@switch: cfg> show config access
{
  "rtbrick-config:access": {
    "l2tp-pool": [
      {
        "pool-name": "lns-pool-example",
        "client-name": "BNG",
        "server-name": "LNS1",
        "client-ipv4": "198.51.100.200",
        "server-ipv4": "198.51.100.219",
        "secret-encrypted-text": "$21e4946e31b406de98b3077aef03ed5a7",
        "preference": 1000,
        "session-limit": 1000
      },
      {
        "pool-name": "lns-pool-example",
        "client-name": "BNG",
        "server-name": "LNS2",
        "client-ipv4": "198.51.100.200",
        "server-ipv4": "198.51.100.220",
        "secret-encrypted-text": "$21e4946e31b406de98b3077aef03ed5a7",
        "preference": 1000,
        "session-limit": 1000
      }
    ]
  }
}
```

Attribute	Description
client-name	Local L2TP tunnel client (LAC) hostname.
server-name	Remote L2TP tunnel server (LNS) hostname.
client-ipv4	Local L2TP tunnel client (LAC) IPv4 address.
server-ipv4	Remote L2TP tunnel server (LNS) IPv4 address.
secret-encrypted-text secret-plain-text	L2TP tunnel secret, which can be provided as plaintext or already encrypted text.
preference	L2TP tunnel preference where the lowest value has the highest priority. **Default:** 0 **Range:** 1 - 65535
session-limit	Tunnels with a session limit reached are not considered for further sessions. This limit has precedence over the default session limit specified in the l2tp-profile. **Default:** 64000 **Range:** 1 - 65535

User Profile Configuration

Subscribers are typically authenticated through remote servers like RADIUS. RBFS additionally offers the capability to authenticate these sessions by comparing them with user profiles defined locally within the system.

Configuring the User Profile

```
supervisor@switch: cfg> set access user-profile
  <user-name>           Username

supervisor@switch: cfg> set access user-profile user@rtbrick.com
  <cr>
  l2tp-pool-name           L2TP pool name
  password-encrypted-text  Secret/password in encrypted text
  password-plain-text      Secret/password in plain text
  tunnel-type              Tunnel type
```

The following example shows a typical ....

```
supervisor@switch: cfg> show config access user-profile user@rtbrick.com
```

```
{
  "rtbrick-config:user-profile": {
    "user-name": "user@rtbrick.com",
    "password-encrypted-text": "$243a1341f44f54888cdd385b9f40513f1",
    "tunnel-type": "PPPoE"
  }
}
```

Attribute	Description
user-name	Username of the subscriber.
password-encrypted-text password-plain-text	User password which can be provided as plaintext or already encrypted text.
tunnel-type	Subscriber tunnel type. **Default:** PPPoE **Values:** PPPoE, L2TP
l2tp-pool-name	Assign a local configured L2TP tunnel pool.

Address Pool Configuration

The way that the address pool configuration relates to all subscriber management configuration tasks is shown in the picture below.

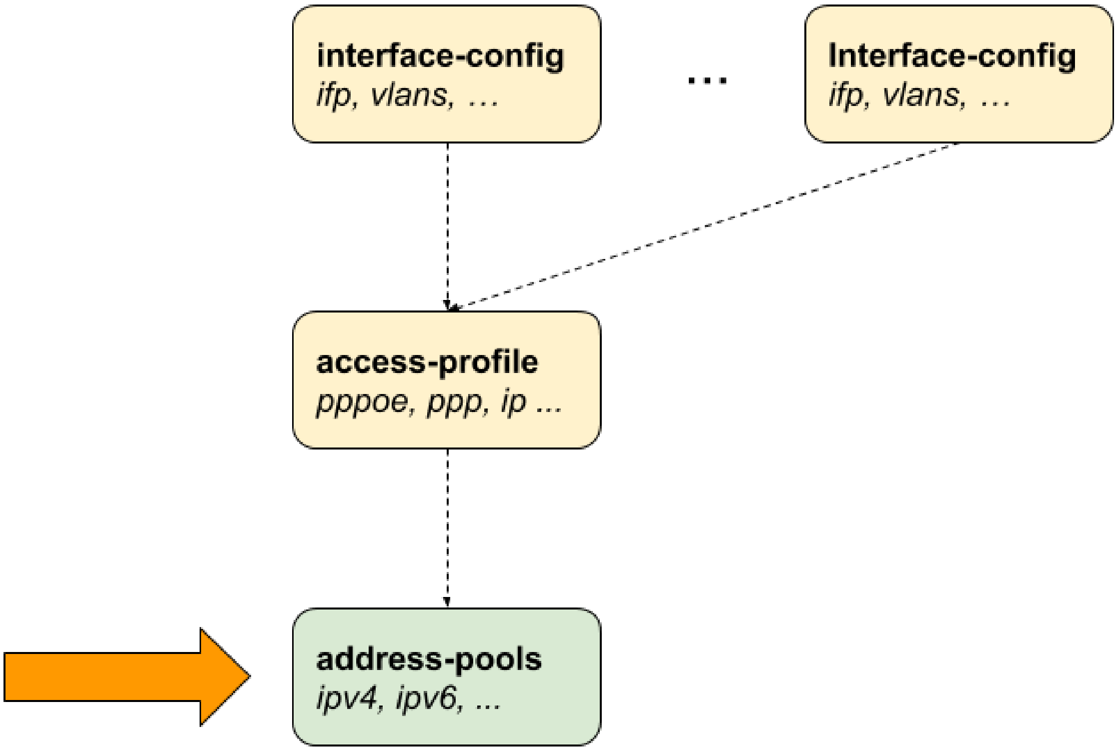


Figure 15. Address Pool Configuration

Configuring the Address Pool

```
   set access pool
  <pool-name>            Name of the address pool

supervisor@switch: cfg> set access pool ipv4-local
  <cr>
  ipv4-address           IPv4 address pool configuration
  ipv6-prefix            IPv6 prefix pool configuration
  next-pool-name         Name of the next address pool to be used if full
```

The following example shows typical IPv4 address and IPv6 prefix pools.

```
supervisor@switch: cfg> show config access
{
  "rtbrick-config:access": {
    "pool": [
      {
        "pool-name": "ipv4-local",
        "ipv4-address": {
          "low": "198.51.100.76",
          "high": "198.51.100.117"
        }
      },
      {
        "pool-name": "ipv6-local",
        "ipv6-prefix": {
          "low": "2001:db8:0:79::/32",
          "high": "2001:db8:0:139::/32"
        }
      },
      {
        "pool-name": "ipv6pd-local",
        "ipv6-prefix": {
          "low": "2001:db8:0:1::/32",
          "high": "2001:db8:0:100::/32"
        }
      }
    ],
  }
}
```

The management of address pools in RBFS offers the flexibility to remove or modify them without causing any disruptions to existing subscribers. This means that even after an address pool has been deleted, the subscribers who were assigned addresses from that pool can still retain and utilize those addresses.

To illustrate, let’s consider a scenario where a subscriber has been allocated an IP address from a specific pool, and later on, that pool is deleted. Despite the pool’s removal, the subscriber can continue using the assigned IP address until their session is terminated. Operators are responsible for ensuring that the address pool is not allocated elsewhere until it’s finally released.

RBFS supports an optional feature known as duplicate address detection. This functionality aims to prevent sessions from logging in if they attempt to use an already-used address. However, it’s important to note that this safeguard doesn’t apply if the address pool is moved to a different BNG. In such cases, duplicate address detection might not work as anticipated, emphasizing the need for careful management and planning when migrating address pools across different BNGs.

Configuring IPv4 Address Pools

```
supervisor@switch: cfg> set access pool ipv4-local ipv4-address
  <cr>
  high                  Highest IPv4 address
  low                   Lowest IPv4 address
  subnet-mask           Subnet mask
```

Attribute	Description
high	Highest IPv4 address.
low	Lowest IPv4 address.

Attribute	Description
subnet-mask	subnet mask allocated to the subscriber.

Configuring IPv6 Prefix Pools

```
supervisor@switch: cfg> set access pool ipv6-local ipv6-prefix
  <cr>
  high                      Highest IPv6 prefix
  low                       Lowest IPv6 prefix
```

Attribute	Description
high	Highest IPv6 prefix.
low	Lowest IPv6 prefix.

ℹ IPv6 prefixes must be at least /64 or larger (/56, /48, …) or /128.

Configuring Linked Pools

Multiple address pools can be linked together using the next-pool-name attribute to form a larger pool of discontinuous ranges.

A - B - C - D

```
supervisor@switch: cfg> set access pool pool-A next-pool pool-B
supervisor@switch: cfg> set access pool pool-B next-pool pool-C
supervisor@switch: cfg> set access pool pool-C next-pool pool-D
```

The actual address pool assigned to a subscriber from the access configuration profile or RADIUS defines the start pool, which could be any pool in the chain. If this pool is already full, the next pool is requested, which repeats until a free pool is found or the end of the chain is reached. RBFS also stops automatically as soon as one pool of the chain was entered twice (loop protection).

This chain can also be closed to a loop to ensure that all pools of a chain are considered if one pool from the middle of the chain is allocated.

A - B - C - D - A

```
supervisor@switch: cfg> set access pool pool-A next-pool pool-B
supervisor@switch: cfg> set access pool pool-B next-pool pool-C
supervisor@switch: cfg> set access pool pool-C next-pool pool-D
```

```
supervisor@switch: cfg> set access pool pool-D next-pool pool-A
```

Control Plane Outbound Marking Configuration

The process of marking all outbound control plane packets is established through the configuration settings found within the forwarding-options, which are illustrated in the following example. This configuration enables effective management and control over the attributes associated with these packets as they traverse the network. By employing these settings, network operators can ensure that control plane traffic is appropriately identified and treated according to specified criteria, contributing to an optimized and well-orchestrated network environment.

```
supervisor@switch: cfg> show config forwarding-options
{
    "rtbrick-config:forwarding-options": {
      "class-of-service": {
        "control-plane-qos": {
          "outbound-marking": {
            "protocol": [
              {
                "protocol": "l2tpv2",
                "remark-type": "tos",
                "codepoint": 64
              },
              {
                "protocol": "ppp",
                "remark-type": "p-bit",
                "codepoint": 6
              },
              {
                "protocol": "radius",
                "remark-type": "tos",
                "codepoint": 64
              }
            ]
          }
        }
      }
    }
  }
```

Configuration Examples

PPPoE

The following example shows a PPPoE configuration for VLAN mode 1:1 with IPv4 and IPv6 enabled, authenticated via RADIUS.

```
{
  "ietf-restconf:data": {
    "rtbrick-config:access": {
      "aaa-profile": [
        {
          "profile-name": "aaa-radius",
          "session-timeout": 0,
          "idle-timeout": 0,
          "aaa-radius-profile": "radius-default",
          "authentication": {
            "order": "RADIUS"
          },
          "accounting": {
            "order": "RADIUS",
          }
        }
      ],
      "radius-profile": [
        {
          "profile-name": "radius-default",
          "nas-identifier": "BNG",
          "nas-port-type": "Ethernet",
          "authentication": {
            "radius-server-profile-name": [
              "radius-server-1",
              "radius-server-2"
              ]
          },
          "accounting": {
            "radius-server-profile-name": [
              "radius-server-1",
              "radius-server-2"
              ],
            "stop-on-reject": "true",
            "stop-on-failure": "true",
            "accounting-on-off": "true",
            "accounting-on-wait": "true",
            "accounting-backup": "true",
            "accounting-backup-max": 86400
          }
        }
      ],
      "radius-server": [
        {
          "server-name": "radius-server-1",
          "address": "198.51.100.101",
          "source-address": "198.51.100.200",
          "secret-encrypted-text": "$21e4946e31b406de98b3077aef03ed5a7",
          "authentication": {
            "enable": "true"
          },
          "accounting": {
            "enable": "true"
          },
          "coa": {
            "enable": "true"
          }
        },
        {
          "server-name": "radius-server-2",
          "address": "198.51.100.102",
          "source-address": "198.51.100.200",
          "secret-encrypted-text": "$21e4946e31b406de98b3077aef03ed5a7",
          "authentication": {
            "enable": "true"
          },
          "accounting": {
            "enable": "true"
          },
          "coa": {
            "enable": "true"
          }
        }
      ],
      "access-profile": [
        {
          "profile-name": "pppoe-dual",
          "protocol": {
            "pppoe": {
              "enable": "true",
              "session-protection": {
                "enable": "true"
              },
              "vlan-priority": 6
            },
            "ppp": {
              "lcp": {
                "authentication-protocol": "PAP_CHAP",
                "echo-interval": 30,
                "echo-max-retransmit": 3,
                "echo-enable": "true"
              },
              "ipcp": {
                "enable": "true",
                "source-ifl": "lo-0/0/0/1"
              },
              "ip6cp": {
                "enable": "true"
              }
            },
            "ra": {
              "enable": "true",
              "interval": 60
            },
            "dhcpv6": {
              "enable": "true"
            },
            "l2tp": {
              "tunnel-profile": "l2tp-default"
            }
          },
          "address-family": {
            "ipv4": {
              "enable": "true",
              "primary-dns": "198.51.100.103",
              "secondary-dns": "198.51.100.104",
              "instance": "default"
            },
            "ipv6": {
              "enable": "true",
              "primary-dns": "2001:db8:0:100::",
              "secondary-dns": "2001:db8:0:104::",
              "instance": "default"
            }
          }
        }
      ],
      "interface": {
        "double-tagged": [
          {
            "interface-name": "ifl-0/0/1",
            "outer-vlan-min": 1,
            "outer-vlan-max": 4094,
            "inner-vlan-min": 7,
            "inner-vlan-max": 7,
            "access-type": "PPPoE",
            "access-profile-name": "pppoe-dual",
            "aaa-profile-name": "aaa-radius"
          }
        ]
      },
      "l2tp-profile": [
        {
          "profile-name": "l2tp-default",
          "session-limit": 4000,
          "client-name": "BNG",
          "client-ipv4": "198.51.100.200",
          "hide-authentication": true
        }
      ]
    },
    "rtbrick-config:interface": [
      {
        "name": "ifl-0/0/1",
        "description": "Access",
        "host-if": "eth0"
      },
      {
        "name": "ifl-0/0/2",
        "description": "Core",
        "host-if": "eth1",
        "unit": [
          {
            "unit-id": 1,
            "address": {
              "ipv4": [
                {
                  "prefix4": "198.51.100.33/24"
                }
              ],
              "ipv6": [
                {
                  "prefix6": "2001:db8:0:32::/32"
                }
              ]
            }
          }
        ]
      },
      {
        "name": "lo-0/0/0",
        "unit": [
          {
            "unit-id": 1,
            "address": {
              "ipv4": [
                {
                  "prefix4": "198.51.100.200/24"
                }
              ]
            }
          }
        ]
      }
    ]
  }
}
```

IPoE

The following example shows an IPoE configuration for VLAN mode 1:1 with IPv4 and IPv6 enabled, authenticated via RADIUS.

```
{
  "ietf-restconf:data": {
    "rtbrick-config:access": {
      "aaa-profile": [
        {
          "profile-name": "aaa-radius",
          "session-timeout": 0,
          "idle-timeout": 0,
          "aaa-radius-profile": "radius-default",
          "authentication": {
            "order": "RADIUS"
          },
          "accounting": {
            "order": "RADIUS",
          }
        }
      ],
      "radius-profile": [
        {
          "profile-name": "radius-default",
          "nas-identifier": "BNG",
          "nas-port-type": "Ethernet",
          "authentication": {
            "radius-server-profile-name": [
              "radius-server-1",
              "radius-server-2"
              ]
          },
          "accounting": {
            "radius-server-profile-name": [
              "radius-server-1",
              "radius-server-2"
              ],
            "stop-on-reject": "true",
            "stop-on-failure": "true",
            "accounting-on-off": "true",
            "accounting-on-wait": "true",
            "accounting-backup": "true",
            "accounting-backup-max": 86400
          }
        }
      ],
      "radius-server": [
        {
          "server-name": "radius-server-1",
          "address": "198.51.100.101",
          "source-address": "198.51.100.200",
          "secret-encrypted-text": "$21e4946e31b406de98b3077aef03ed5a7",
          "authentication": {
            "enable": "true"
          },
          "accounting": {
            "enable": "true"
          },
          "coa": {
            "enable": "true"
          }
        },
        {
          "server-name": "radius-server-2",
          "address": "198.51.100.102",
          "source-address": "198.51.100.200",
          "secret-encrypted-text": "$21e4946e31b406de98b3077aef03ed5a7",
          "authentication": {
            "enable": "true"
          },
          "accounting": {
            "enable": "true"
          },
          "coa": {
            "enable": "true"
          }
        }
      ],
      "access-profile": [
        {
          "profile-name": "ipoe-dual",
          "protocol": {
            "dhcp": {
              "enable": "true",
              "mode": "server"
            },
            "dhcpv6": {
              "enable": "true",
              "mode": "server"
            },
          },
          "address-family": {
            "ipv4": {
              "enable": "true",
              "pool-name":"ipoe",
              "primary-dns": "198.51.100.103",
              "secondary-dns": "198.51.100.104",
              "instance": "default"
            },
            "ipv6": {
              "enable": "true",
              "pool-name":"ipoe-ia-na",
              "prefix-delegation-pool-name":"ipoe-ia-pd",
              "primary-dns": "2001:db8:0:100::",
              "secondary-dns": "2001:db8:0:104::",
              "instance": "default"
            }
          }
        }
      ],
      "interface": {
        "double-tagged": [
          {
            "interface-name": "ifl-0/0/1",
            "outer-vlan-min": 1,
            "outer-vlan-max": 4094,
            "inner-vlan-min": 7,
            "inner-vlan-max": 7,
            "access-type": "IPoE",
            "access-profile-name": "ipoe-dual",
            "aaa-profile-name": "aaa-radius",
            "gateway-ifl": "lo-0/0/0/1"
          }
        ]
      },
      "pool": [
        {
          "pool-name": "ipoe",
          "ipv4-address": {
            "low": "10.0.0.1",
            "high": "10.0.255.255"
          }
        },
        {
          "pool-name": "ipoe-ia-na",
          "ipv6-prefix": {
            "low": "fc66::1/128",
            "high": "fc66::ffff/128"
          }
        },
        {
          "pool-name": "ipoe-ia-pd",
          "ipv6-prefix": {
            "low": "fc66:0:100::/56",
            "high": "fc66:0:1ff:ff00::/56"
          }
        }
      ],
    },
    "rtbrick-config:interface": [
      {
        "name": "ifl-0/0/1",
        "description": "Access",
        "host-if": "eth0"
      },
      {
        "name": "ifl-0/0/2",
        "description": "Core",
        "host-if": "eth1",
        "unit": [
          {
            "unit-id": 1,
            "address": {
              "ipv4": [
                {
                  "prefix4": "198.51.100.33/24"
                }
              ],
              "ipv6": [
                {
                  "prefix6": "2001:db8:0:6423::/32"
                }
              ]
            }
          }
        ]
      },
      {
        "name": "lo-0/0/0",
        "unit": [
          {
            "unit-id": 1,
            "address": {
              "ipv4": [
                {
                  "prefix4": "198.51.100.200/24"
                }
              ]
            }
          }
        ]
      }
    ]
  }
}
```

4.1.3. Operations

Subscriber Management

The following commands are served by subscriber daemon and are applicable for all kinds of subscribers like PPPoE, L2TP or IPoE.

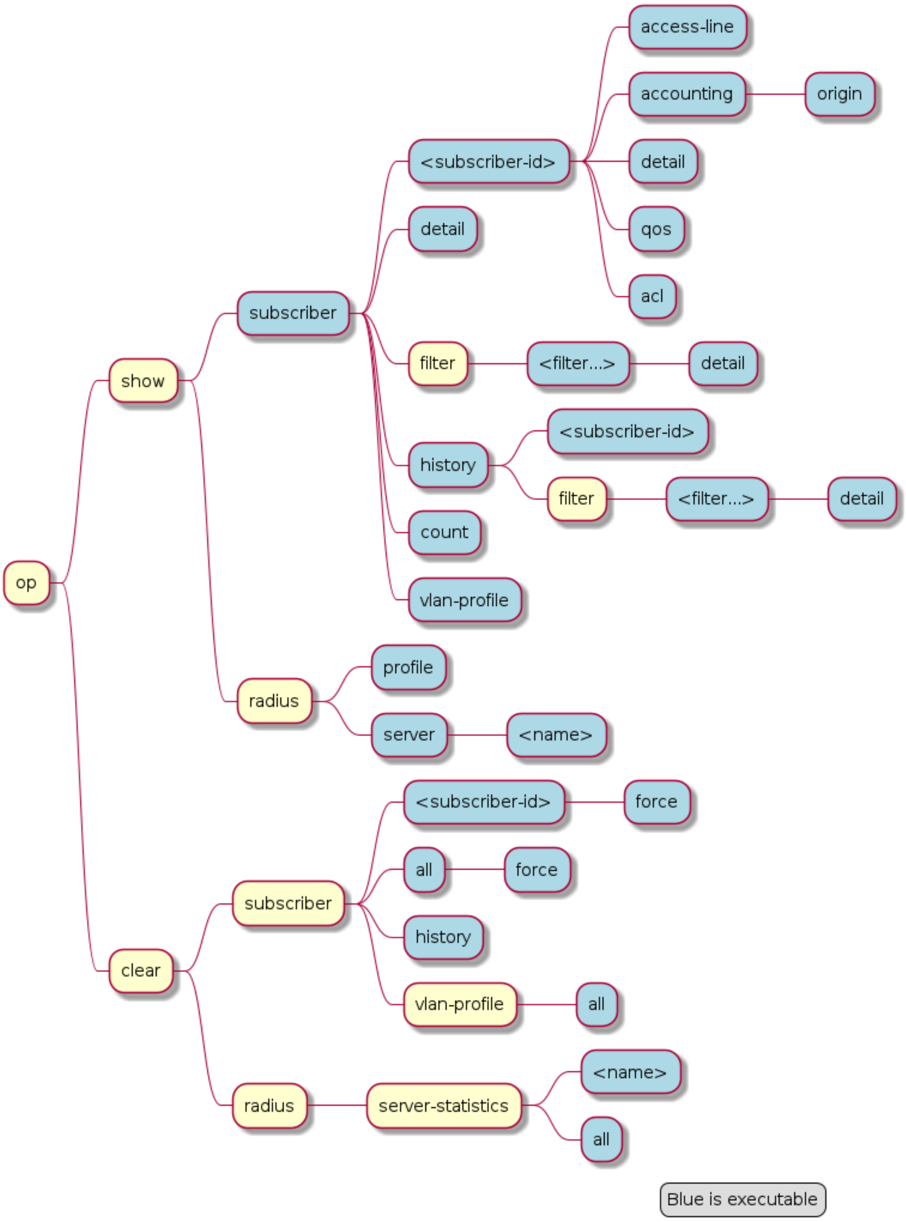


Figure 16. Subscriber Management Operational Commands

Subscribers

The term subscriber describes an access user or session from a higher level decoupled from underlying protocols like PPPoE or IPoE. Subscribers in RBFS can be managed locally or remote via RADIUS. Each subscriber is uniquely identified by a 64bit number called subscriber-id.

Subscriber States

A good starting point for troubleshooting subscriber services is to verify the status of the subscriber sessions. The state ESTABLISHED means that the session is fully operational.

```
supervisor@leaf1: op> show subscriber
Subscriber-Id          Interface        VLAN      Type   State
72339069014638600      ifp-0/0/1        1:1       PPPoE  ESTABLISHED
72339069014638601      ifp-0/0/1        1:2       PPPoE  ESTABLISHED
72339069014638602      ifp-0/0/1        1:3       PPPoE  ESTABLISHED
72339069014638603      ifp-0/0/3        2000:7    L2TP   ESTABLISHED
```

Alternative use show subscriber detail which shows further details like username, Agent-Remote-Id (aka Line-Id) or Agent-Circuit-Id if screen width is large enough to print all those information.

The meaning of the subscriber state is shown in the following table and diagram.

State	Description
INIT	Initial subscriber state.
AUTHENTICATING	Authenticate the subscriber using the configured method.
ADDRESS ALLOCATION	Allocate (RADIUS or pool) and validate (DAD) addresses.
TUNNEL SETUP	Setup tunnel resources (L2TP or L2X).
IFL SETUP	Create subscriber IFL with corresponding QoS resources.
FULL	Wait for subscriber to be in forwarding state. Inform underlying protocols (PPPoED or IPoED) to continue with session setup.
ACCOUNTING	Start subscriber accounting and wait for response.
ESTABLISHED	The subscriber becomes ESTABLISHED after response to RADIUS Accounting-Request-Start if RADIUS accounting is enabled otherwise immediately after FULL.
TERMINATING	The subscriber remains in this state until all resources are freed and accounting stopped. This means that subscriber remain in this state until response to RADIUS Accounting-Request-Stop if RADIUS accounting is enabled.

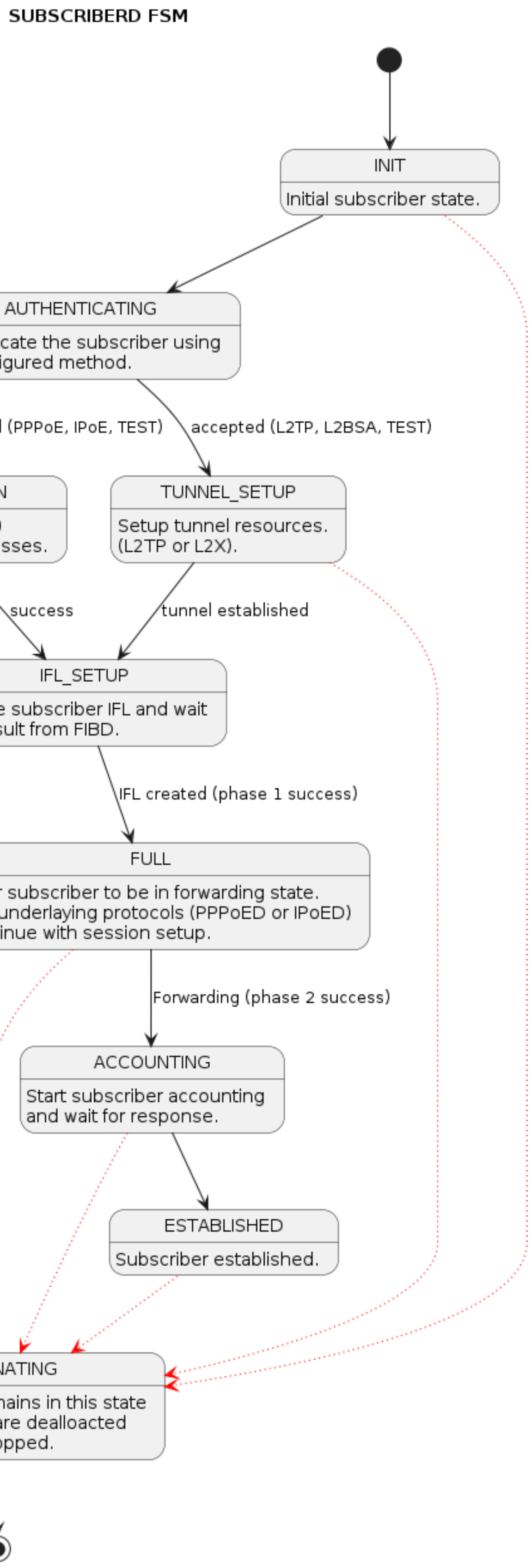

SUBSCRIBERD FSM
INIT
Initial subscriber state.
AUTHENTICATING
Authenticate the subscriber using the configured method.
accepted (PPPoE, IPoE, TEST)
accepted (L2TP, L2BSA, TEST)
ADDR_ALLOCATION
Allocate (RADIUS or pool) and validate (DAD) addresses.
TUNNEL_SETUP
Setup tunnel resources. (L2TP or L2X).
success
tunnel established
IFL_SETUP
Create subscriber IFL and wait for result from FIBD.
IFL created (phase 1 success)
FULL
Wait for subscriber to be in forwarding state. Inform underlaying protocols (PPPoED or IPoED) to continue with session setup.
Forwarding (phase 2 success)
ACCOUNTING
Start subscriber accounting and wait for response.
ESTABLISHED
Subscriber established.
TERMINATING
The subscriber remains in this state until all resources are dealloacted and accounting stopped.

Figure 17. Subscriber States

For each subscriber a set of commands is available showing detailed information.

```
supervisor@leaf1: op> show subscriber 72339069014638594
  <cr>
  access-line           Subscriber access line information
  accounting            Subscriber accounting information
  acl                   Subscriber ACL information (filter)
  detail                Detailed subscriber information
  qos                   Subscriber QoS information

user@switch: op> show subscriber 72339069014638594 detail
Subscriber-Id: 72339069014638594
    Type: PPPoE
    State: ESTABLISHED
    Created: Fri Sep 18 20:50:02 GMT +0000 2020
    Interface: ifl-0/0/1
    Outer VLAN: 128
    Inner VLAN: 7
    Client MAC: fe:08:e8:ea:1d:32
    Server MAC: 7a:52:4a:01:00:01
    IFL: ppp-0/0/1/72339069014638594
    Username: 1122334455#123456789#0001@t-online.de
    Agent-Remote-Id: DEU.DTAG.1337
    Agent-Circuit-Id: 0.0.0.0/0.0.0.0 eth 1337
    Access-Profile: access-profile1
    AAA-Profile: aaa-profile1
    Session-Timeout: 30000
    Idle-Timeout: 120
    IPv4:
        Instance: default
        Address: 198.51.100.116/255.255.255.255
        Address Active: True
        Primary DNS: 198.51.100.213
        Secondary DNS: 198.51.100.54
    IPv6:
        Instance: default
        RA Prefix: 2001:db8:0:400::/32
        RA Prefix Active: True
        Delegated Prefix (DHCPv6): 2001:db8:0:269::/56
        Delegated Prefix Active: False
        Primary DNS: 2001:db8:0:92::
        Secondary DNS: 2001:db8:0:174::
    Accounting:
        Session-Id: 72339069014638594:1600462202
        Start-Time: 2020-09-18T20:50:02.738306+0000
        Interims Interval: 30 seconds
```

Subscriber Termination Codes

The following command shows the reasons why subscribers are terminated for the last 24 hours and up to 4000 subscribers.

```
supervisor@leaf1: op> show subscriber history
Subscriber-Id          Timestamp                            Terminate Code
72339069014638594      Fri Oct 16 20:17:33 GMT +0000 2020   Accounting-Request-On
Wait
72339069014638595      Fri Oct 16 20:32:19 GMT +0000 2020   PPPoE LCP Terminate
Request Received
```

Subscriber Count

To view a summary of PPPoE, L2TP, IPoE, and L2BSA subscribers in the Setup, Established, and Terminating state, use the "show subscriber count" command. This command provides information per interface and a total summary based on subscriber type.

```
supervisor@leaf1: op> show subscriber count
                       Total        Setup        Established  Terminating
Summary                18000        0            18000        0
  PPPoE                18000        0            18000        0
  L2TP                 0            0            0            0
  IPoE                 0            0            0            0
  L2BSA                0            0            0            0
ifp-0/1/30             6000         0            6000         0
  PPPoE                6000         0            6000         0
  L2TP                 0            0            0            0
  IPoE                 0            0            0            0
  L2BSA                0            0            0            0
ifp-0/1/32             6000         0            6000         0
  PPPoE                6000         0            6000         0
  L2TP                 0            0            0            0
  IPoE                 0            0            0            0
  L2BSA                0            0            0            0
ifp-0/1/33             6000         0            6000         0
  PPPoE                6000         0            6000         0
  L2TP                 0            0            0            0
  IPoE                 0            0            0            0
  L2BSA                0            0            0            0
supervisor@leaf1: op>
```

RADIUS

RADIUS Profile

The following command shows the status of all RADIUS profiles.

```
supervisor@leaf1: op> show radius profile
RADIUS Profile: radius-default
    NAS-Identifier: BNG
    NAS-Port-Type: Ethernet
    Authentication:
        Algorithm: ROUND-ROBIN
        Server:
            radius-server-1
            radius-server-2
    Accounting:
        State: UP
        Stop on Reject: True
        Stop on Failure: True
        Backup: True
        Algorithm: ROUND-ROBIN
        Server:
            radius-server-1
            radius-server-2
```

This meaning of the accounting state is explained in the table below.

Code	State	Description
0x00	DISABLED	Change profile accounting state from DISABLED to ACTIVE if at least one server referenced is found with accounting enabled.
0x01	ACTIVE	Server referenced by RADIUS profile but no response received
0x02	STARTING	Send accounting-on and wait for response.
0x05	UP	Change profile accounting state to UP if at least one referenced accounting server is UP.

The profile state becomes immediately ACTIVE if at least one of the referenced accounting servers can be found in RADIUS server table with accounting enabled. Otherwise the profile keeps DISABLED.

If RADIUS Accounting-On is enabled, the profile state becomes STARTING before UP. It is not permitted to send any accounting request start, interim or stop related to a profile in this state. It is also not permitted to send authentication requests if **accounting-on-wait** is configured in addition. The state becomes UP if at least one server in the accounting server list is in a state UP or higher (UNREACHABLE, DOWN, TESTING, DEAD).

A new profile added which references existing used RADIUS servers must not trigger a RADIUS Accounting-On request if at least one of the referenced servers is in a state of UP or higher.

RADIUS Server

The following command shows the status of all RADIUS servers.

```
supervisor@leaf1: op> show radius server
RADIUS Server            Address          Authentication State Accounting State
radius-server-1          198.51.100.64    ACTIVE               UP
radius-server-2          198.51.100.163   ACTIVE               ACTIVE
radius-server-3          198.51.100.104   ACTIVE               ACTIVE
```

This meaning of those states is explained in the table and diagram below.

Code	State	Description
0x00	DISABLED	RADIUS authentication (authentication state) or accounting (accounting state) is disabled or server not referenced by profile.
0x01	ACTIVE	Server referenced by RADIUS profile but no valid response received.
0x02	STARTING	This state is valid for accounting (accounting state) only during accounting-on is sending (wait for accounting-on response).
0x03	STOPPING	This state is valid for accounting (accounting state) only during accounting-off is sending (wait for accounting-off response).
0x04	FAILED	This state is valid for accounting (accounting state) only if accounting-on/off timeout occurs.
0x05	UP	Valid RADIUS response received
0x06	UNREACHABLE	No response received/timeout but server is still usable.
0x07	DOWN	Server is down but can be selected.
0x08	TESTING	Send a request to test if server is back again. The server will not be selected for another request in this state (use a single request to check if server is back again).
0x09	DEAD	Server is down and should not be selected.

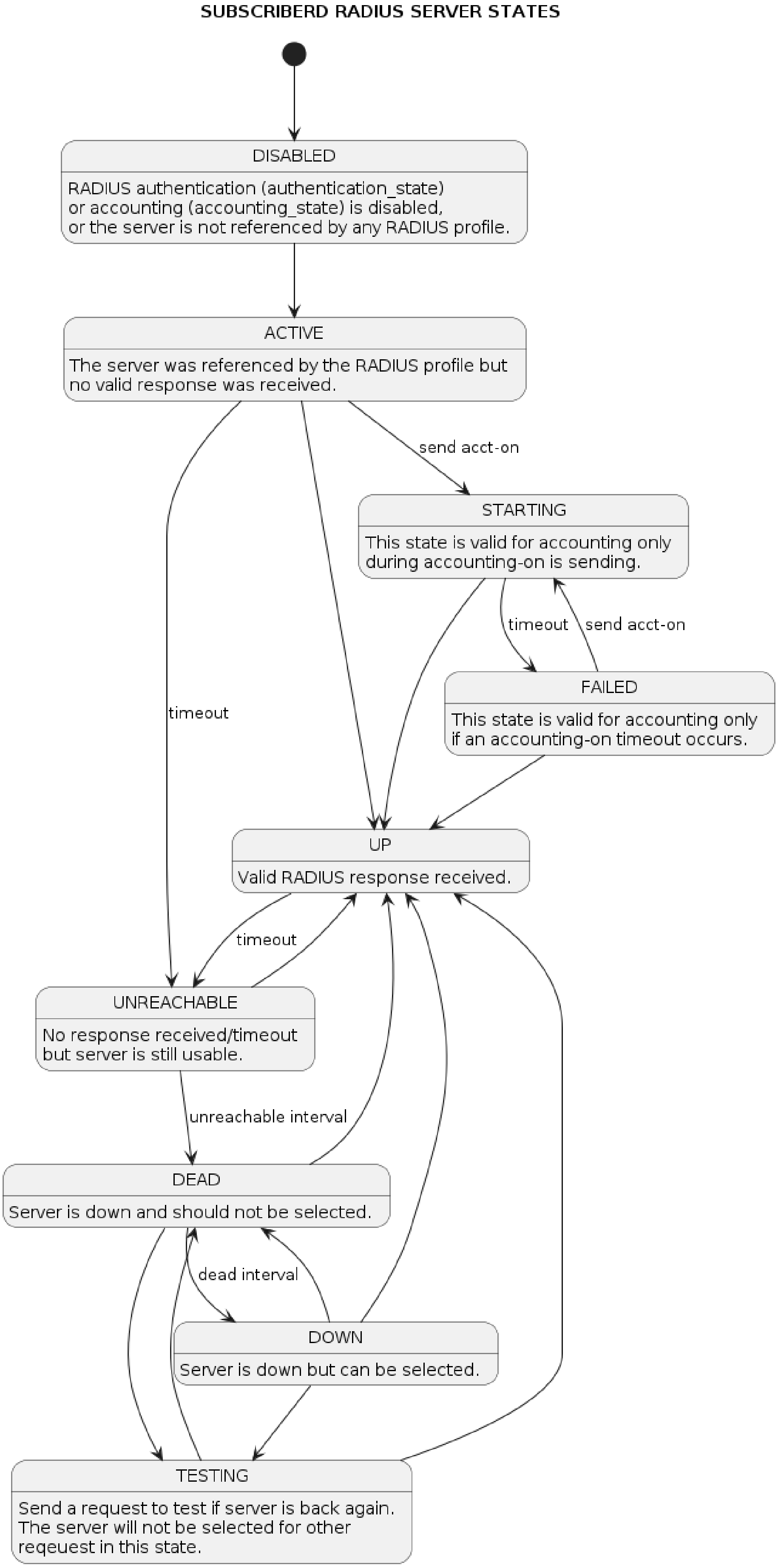

SUBSCRIBERD RADIUS SERVER STATES
DISABLED
RADIUS authentication (authentication_state)
or accounting (accounting_state) is disabled,
or the server is not referenced by any RADIUS profile.
ACTIVE
The server was referenced by the RADIUS profile but
no valid response was received.
send acct-on
STARTING
This state is valid for accounting only
during accounting-on is sending.
timeout
send acct-on
FAILED
This state is valid for accounting only
if an accounting-on timeout occurs.
timeout
UP
Valid RADIUS response received.
timeout
UNREACHABLE
No response received/timeout
but server is still usable.
unreachable interval
DEAD
Server is down and should not be selected.
dead interval
DOWN
Server is down but can be selected.
TESTING
Send a request to test if server is back again.
The server will not be selected for other
reqeuest in this state.

Figure 18. RADIUS Server States

For each server dedicated detailed information are displayed with the following commands.

```
supervisor@leaf1: op> show radius server radius-server-1
RADIUS Server: radius-server-1
    Address: 198.51.100.64
    Source: 198.51.100.200
    Rate: 600 PPS
    Rate Tokens: 600
    Dropped: 0
    Authentication:
        State: ACTIVE
        State Changed: Fri Oct 16 20:17:27 GMT +0000 2020
        Port: 1812
        Retry: 3
        Timeout: 5
        Outstanding: 100
        Statistics:
            Request Sent: 0
            Request Retry: 0
            Request Timeout: 0
            Accept Received: 0
            Reject Received: 0
            Dropped: 0
    Accounting:
        State: UP
        State Changed: Fri Oct 16 20:18:27 GMT +0000 2020
        Port: 1813
        Retry: 10
        Timeout: 30
        Outstanding: 100
        Statistics:
            Request Sent: 1
            Request Retry: 2
            Request Timeout: 0
            Response Received: 1
            Dropped: 0
    CoA:
        Port: 3799
        Statistics:
            Request Received: 0
            Dropped: 0
```

PPPoE

The following commands are applicable for PPPoE sessions only.

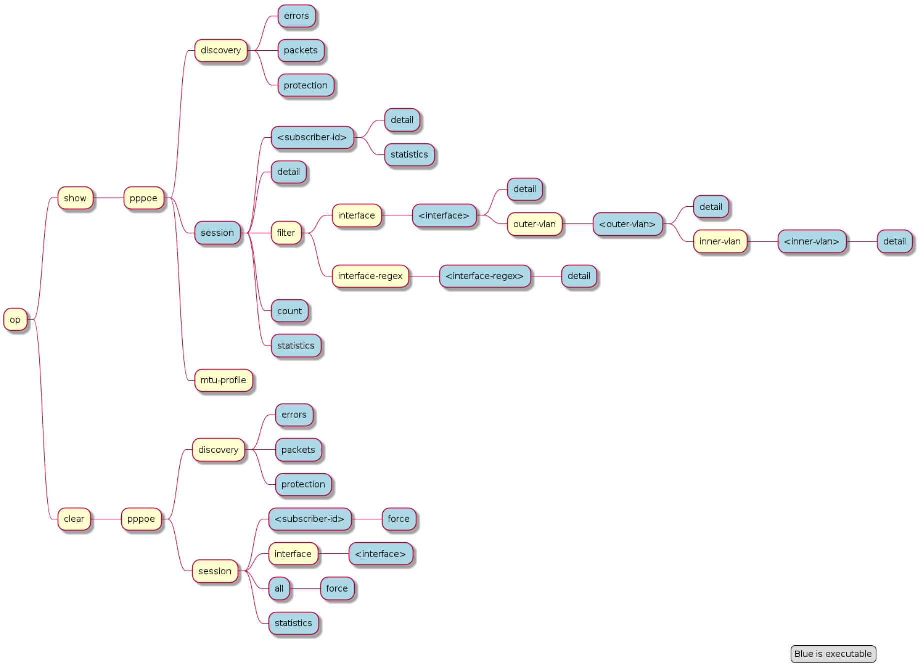


Figure 19. PPPoE Operational Commands

For PPPoE sessions the state should be ESTABLISHED if local terminated or TUNNELLED for L2TPv2 tunnelled sessions.

```
supervisor@rtbrick: op> show pppoe session
Subscriber-Id          Interface        VLAN      MAC               State
72339069014638604      ifp-0/0/1        1:1       00:04:0e:00:00:01 ESTABLISHED
72339069014638601      ifp-0/0/1        1:2       00:04:0e:00:00:02 ESTABLISHED
72339069014638602      ifp-0/0/1        1:3       00:04:0e:00:00:03 ESTABLISHED
72339069014638603      ifp-0/0/3        2000:7    52:54:00:57:c8:29 TUNNELLED
```

Alternative use show pppoe session detail which shows further details like username, Agent-Remote-Id (aka Line-Id) or Agent-Circuit-Id if screen width is large enough to print all those information.

State	Description
LINKING	PPP LCP setup.
AUTHENTICATING	PPP authentication (PAP or CHAP).
NETWORKING	PPP IPCP (IPv4) and IP6CP (IPv6) setup.
ESTABLISHED	The PPPoE session becomes established if at least one NCP (IPCP or IP6CP) is established (state OPEN).
TUNNELLED	This state indicates that a PPPoE session is tunnelled via L2TPv2.
TERMINATING	PPP session teardown.

State	Description
TERMINATED	PPPoE session terminated.

If PPPoE session remain in state TERMINATED, the subscriber state should be checked. Typically this happens if RADIUS Accounting-Request-Stop is still pending.

Further details per PPPoE session can be shown with the following commands.

```
supervisor@rtbrick: op> show pppoe session 72339069014638648
  <cr>
  detail                Detailed session information
  statistics            Protocol statistics
```

The detail command shows the states of the session and all sub-protocols with extensive information and negotiated parameters.

```
user@switch: op> show pppoe session 72339069014638648 detail
Subscriber-Id: 72339069014638648
    State: ESTABLISHED
    Uptime: Tue Nov 17 11:46:43 GMT +0000 2020 (0:00:21.979775)
    Interface: ifp-0/0/3
    Outer VLAN: 10
    Inner VLAN: 7
    Client MAC: 52:54:00:57:c8:29
    Server MAC: 7a:52:4a:c0:00:03
    Session-Id: 55
    Host-Unique: 00000001
    Agent-Remote-Id: DEU.RTBRICK.1
    Agent-Circuit-Id: 0.0.0.0/0.0.0.0 eth 1
    Access-Profile: pppoe-dual
    AAA-Profile: aaa-default
    PPP LCP:
        State: OPENED
        Negotiated Protocols: CHAP, IPCP, IP6CP
        Negotiated Parameters: MRU, AUTH, MAGIC
        Magic Number: 1079931229 Peer: 3432759752
        MRU: 1492 Peer: 1492
        MTU: 1492 Profile: __default_pppoe__
        Echo Interval: 30 seconds
    CHAP Authentication:
        State: COMPLETED
        Username: user1@rtbrick.com
    PPP IPCP:
        State: OPENED
        Instance: default
        IP Address: 198.51.100.200 Peer: 198.51.100.72
        Primary DNS: 198.51.100.88
        Secondary DNS: 198.51.100.54
    PPP IP6CP:
        State: OPENED
        Instance: default
        Interface Identifier: c5f6:1dbd:8cc1:bea9
        Peer Interface Identifier: 5054:00ff:fe57:c829
    IPv6:
        RA Interval: 60 seconds
        RA Prefix: 2001:db8:0:246::/32
        Delegated Prefix (DHCPv6): 2001:db8:0:9::/32 Assigned: True
        Primary DNS: 2001:db8:0:114::
        Secondary DNS: 2001:db8:0:115::
    Control Traffic Statistics:
        Ingress: 15 packets 1059 bytes
        Egress: 16 packets 1475 bytes
```

Session statistics are available global and per session.

```
supervisor@rtbrick: op> show pppoe session statistics
supervisor@rtbrick: op> show pppoe session 72339069014638601 statistics
```

The PPPoE discovery statistics are helpful if session setup fails in initial PPPoE tunnel setup before actual PPP negotiation is starting.

```
supervisor@rtbrick: op> show pppoe discovery packets
Packet           Received         Sent
PADI             17               0
PADO             0                17
PADR             17               0
PADS             0                17
PADT             1                13

supervisor@rtbrick: op> show pppoe discovery errors
PADI Drop No Config            : 0
PADI Drop Session Protection   : 0
PADI Drop Session Limit        : 0
PADI Drop Dup Session          : 0
PADI Drop Interface Down       : 0
PADR Drop No Config            : 0
PADR Drop Wrong MAC            : 0
PADR Drop Interface Down       : 0
PADR Drop Session Limit        : 0
PADR Drop Session Protection   : 0
PADR Drop Bad Cookie           : 0
PADR Drop Bad Session          : 0
PADR Drop Dup Session          : 0
PADR Drop No mapping Id        : 0
PADT Drop No Session           : 0
PADT Drop Wrong MAC            : 0
PADX Interface Get Failure     : 0
```

If PPPoE session protection is enabled in access configuration profile, short lived or failed sessions will be logged in the PPPoE session protection table (local.pppoe.session.protection).

Every session not established for at least 60 seconds per default is considered as failed or short lived session. This will block new sessions on this IFP and VLAN's for one second per default which increase exponential with any further failed session until the max time of per default 300 seconds is reached. The interval is reset after 900 seconds without failed sessions.

The PPPoE session protection table include also last subscriber-id and terminate code which indicates the reason for session failures.

```
supervisor@rtbrick: op> show pppoe discovery protection
Interface        VLAN      Status  Attempts   Last Terminate Code
ifp-0/0/1        1:1       OK      1          PPPoE LCP Terminate Request Received
ifp-0/0/1        1:2       OK      1          PPPoE LCP Terminate Request Received
ifp-0/0/1        1:3       OK      1          PPPoE LCP Terminate Request Received
```

If status OK indicates that new session are accepted where BLOCKED means that sessions will be rejected.

L2TP

The following commands are applicable for L2TP only.

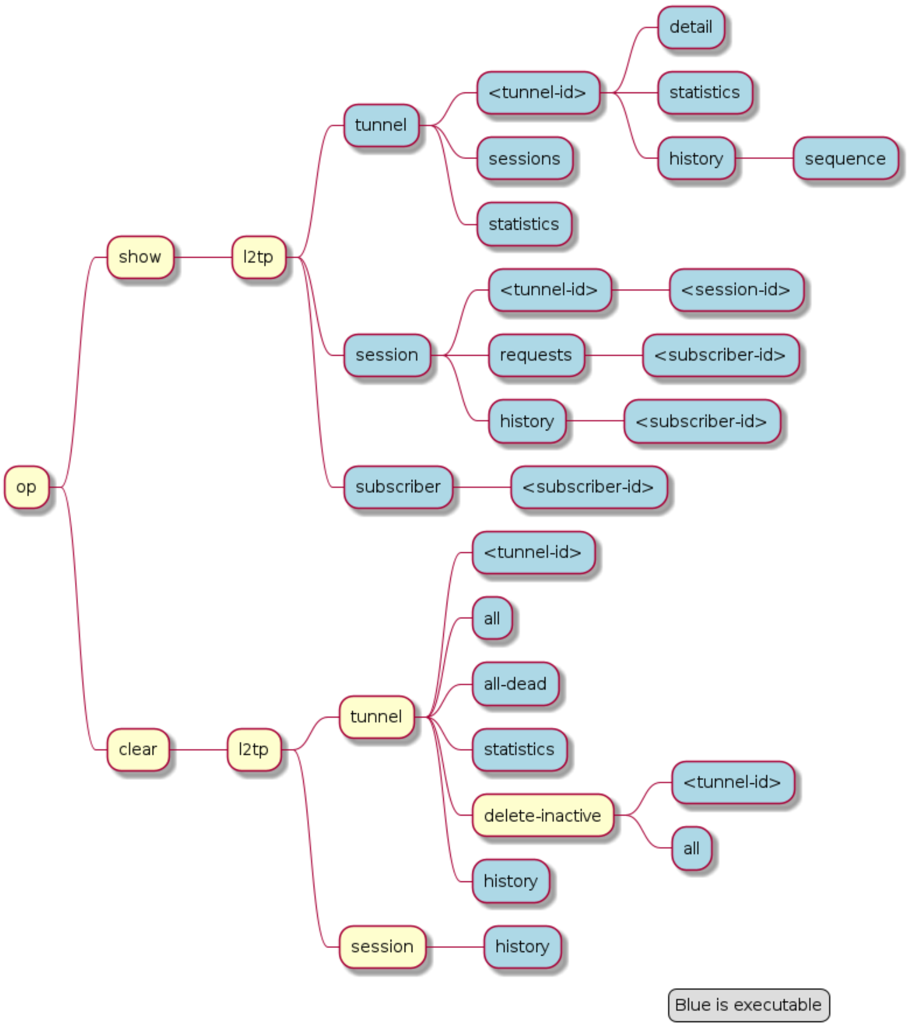


Figure 20. L2TP Operational Commands

For L2TPv2 tunnelled PPPoE sessions the global unique subscriber-id can be used to get information about the L2TP session.

```
supervisor@rtbrick: op> show l2tp subscriber 72339069014638621
Subscriber-Id: 72339069014638621
    State: ESTABLISHED
    Local TID: 45880
    Local SID: 39503
    Peer TID: 1
    Peer SID: 1
    Call Serial Number: 10
    TX Speed: 10007000 bps
    RX Speed: 1007000 bps
    CSUN: disabled
```

The following command gives a good overview over the corresponding tunnels.

```
supervisor@leaf1: op> show l2tp tunnel sessions
Role Local TID Peer TID State        Preference Sessions Established Peer Name
LAC        2022        1 ESTABLISHED       10000        1           1 LNS3
LAC       3274        1 ESTABLISHED       10000        1           1 LNS8
LAC      14690        1 ESTABLISHED       10000        1           1 LNS6
LAC      29489        1 ESTABLISHED       10000        1           1 LNS9
LAC      33323        1 ESTABLISHED       10000        1           1 LNS4
LAC      35657        1 ESTABLISHED       10000        1           1 LNS10
LAC      37975        1 ESTABLISHED       10000        1           1 LNS1
LAC      45880        1 ESTABLISHED       10000        1           1 LNS7
LAC      46559        1 ESTABLISHED       10000        1           1 LNS2
LAC      58154        1 ESTABLISHED       10000        1           1 LNS5
```

Detailed information per tunnel are available via show l2tp tunnel <TID> detail.

L2TP tunnel statistics are available global and per tunnel.

```
supervisor@leaf1: op> show l2tp tunnel statistics
supervisor@leaf1: op> show l2tp tunnel 37975 statistics
```

L2TP Result and Disconnect Codes

The received result (RFC2661) and disconnect (RFC3145) code and message from CDN and StopCCN will be stored similar to the subscriber terminate history table for 24 hours and up to 1000 records.

```
supervisor@leaf1: op> show l2tp tunnel history
Sequence Local TID Peer TID Timestamp                            Terminate Code
       1     34209        0 Wed Jul 28 13:02:35 GMT +0000 2021   Admin Request
       2     39860        1 Wed Jul 28 13:02:35 GMT +0000 2021   Admin Request
       3     39860        2 Wed Jul 28 13:02:54 GMT +0000 2021   Admin Request
       4     39860        3 Wed Jul 28 13:04:29 GMT +0000 2021   StopCCN Received
(Requester is being shut down)
       5     39860        1 Wed Jul 28 13:06:19 GMT +0000 2021   StopCCN Received
(Requester is being shut down)

supervisor@leaf1: op> show l2tp tunnel history 4
Local TID: 39860 Peer TID: 3
    Terminate Code: StopCCN Received
    Timestamp: Wed Jul 28 13:04:29 GMT +0000 2021
    Local Address: 198.51.100.102
    Peer Address: 198.51.100.133
    Peer Name: LNS1
    Tunnel-Client-Auth-ID: BNG
    Tunnel-Server-Auth-ID: LNS1
    Result Code: Requester is being shut down

supervisor@leaf1: op> show l2tp session history
Subscriber-Id          Local TID Local SID Terminate Code
72339069014638614          39860      5597 Clear Session
72339069014638615          39860      5208 Clear Session
72339069014638623          39860     29626 Clear Session
72339069014638624          39860     42480 L2TP Tunnel Down
72339069014638625          39860     34417 L2TP Tunnel Down
72339069014638626          39860     20229 L2TP Tunnel Down
```

The show subscriber history <subscriber-id> command will also return L2TP details if found for the corresponding subscriber.

```
supervisor@leaf1: op> show subscriber history 72339069014638703
Subscriber-Id: 72339069014638703
    Terminate Code: L2TP CDN Request
    Timestamp: Wed Jul 28 13:06:18 GMT +0000 2021
    Interface: ifl-0/0/1
    Outer VLAN: 1000
    Inner VLAN: 2002
    Client MAC: 02:00:00:00:00:04
    Username: blaster@l2tp.de
    Agent-Remote-Id: DEU.RTBRICK.2
    Agent-Circuit-Id: 0.0.0.0/0.0.0.0 eth 0:2
    Accounting-Session-Id: 72339069014638703:1627477569
    L2TP Disconnect Cause:
        Code: Normal disconnection (LCP terminate-request)
        Protocol: 0
        Direction: Peer
        Message: N/A
```

IPoE

The following commands are applicable for IPoE subscribers only.

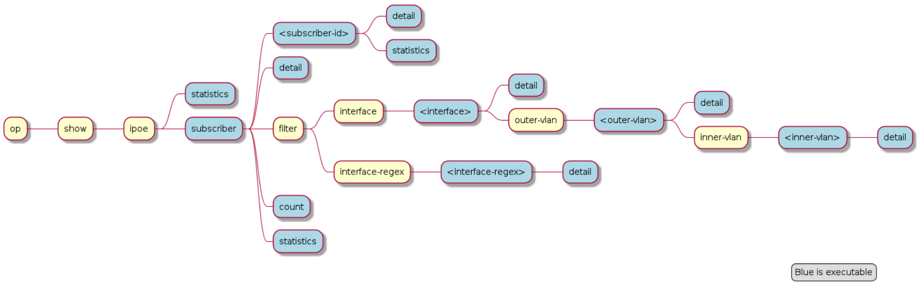


Figure 21. IPoE Operational Commands

```
supervisor@leaf1: op> show ipoe subscriber detail
Subscriber-Id          Interface        VLAN      MAC               State
DHCPv4     DHCPv6
216454257090494465    ifl-0/0/1     8:1       02:00:00:00:00:01 ESTABLISHED
Bound      Bound
216454257090494466    ifl-0/0/1     8:2       02:00:00:00:00:02 ESTABLISHED
Bound      Bound
216454257090494467    ifl-0/0/1     8:3       02:00:00:00:00:03 ESTABLISHED
Bound      Bound
216454257090494468    ifl-0/0/1     8:4       02:00:00:00:00:04 ESTABLISHED
Bound      Bound
```

Further details per subscriber can be shown with the following command.

```
supervisor@leaf1: op> show ipoe subscriber 216454257090494465 detail
```

```
Subscriber-Id: 216454257090494465
    State: ESTABLISHED
    Uptime: Mon Jun 14 15:46:15 GMT +0000 2021 (0:02:19.421591)
    Interface: ifl-0/0/1
    Outer VLAN: 8
    Inner VLAN: 1
    Client MAC: 02:00:00:00:00:01
    Gateway Interface: lo-0/0/0/1
    Gateway Instance: default
    Gateway IPv4: 198.51.100.200/255.255.255.255
    Gateway MAC: 7a:52:4a:c0:00:01
    Agent-Remote-Id: DEU.RTBRICK.1
    Agent-Circuit-Id: 0.0.0.0/0.0.0.0 eth 0:1
    DHCPv4:
        Mode: Server
        State: Bound
        Address: 198.51.100.202/255.255.255.255
        Lease Created: Mon Jun 14 15:46:15 GMT +0000 2021 (0:02:19.427443)
        Lease Time: 300 seconds
        Lease Expire: 161 seconds
    DHCPv6:
        Mode: Server
        State: Bound
        Client DUID: 00030001020000000001
        Server DUID: 0003001b78524afffec00001
        IA_NA:
            Address: 2001:db8:0:96
            IAID: 1181407340
            Active: True
        IA_PD:
            Prefix: 2001:db8:0:333/32
            IAID: 4095128883
            Active: True
        Lease Created: Mon Jun 14 15:46:15 GMT +0000 2021 (0:02:19.428676)
        Lease Time (Lifetime): 14400 seconds
        Lease Expire: 14261 seconds
        Preferred Lifetime: 1800 seconds
```

Local Address Pools

> Rather than using recommended IP addresses for technical documents, the document shows actual IP pool ranges.

The usage of local address pools can be monitored using the show subscriber pool commands as shown below.

```
supervisor@switch: op> show subscriber pool summary
Pool Name                        AFI  Usage           Range
pool-A                           IPv4 256/256         10.0.1.0 - 10.0.1.255
pool-B                           IPv4 2/256           10.0.2.0 - 10.0.2.255
pool-C                           IPv4 0/256           10.0.3.0 - 10.0.3.255
pool-D                           IPv4 0/256           10.0.4.0 - 10.0.4.255

supervisor@switch: op> show subscriber pool ipv4 pool-A
Pool Name: pool-A
    AFI: IPv4
    Usage: 256/256
    Range: 10.0.1.0 - 10.0.1.255
    Next: pool-B

supervisor@switch: op> show subscriber pool ipv4 pool-B
Pool Name: pool-B
    AFI: IPv4
    Usage: 2/256
    Range: 10.0.2.0 - 10.0.2.255
    Next: pool-C

supervisor@switch: op> show subscriber pool ipv4 pool-B allocation
Subscriber-Id          Timestamp                            Address/Prefix
72339069014638598      Wed Sep 15 09:02:15 GMT +0000 2021   10.0.2.0
72339069014638602      Wed Sep 15 09:02:15 GMT +0000 2021   10.0.2.1
```

4.1.4. Supported Standards

ℹ RFC and draft compliance are partial except as specified.

PPPoE

- RFC 1516
- RFC 1661 (partly)
- RFC 1332 (partly)
- RFC 5072 (partly)
- RFC 1334 (partly)

RADIUS

- RFC 2865 (partly)
- RFC 3162 (partly)
- RFC 2866 (partly)
- RFC 4372 (partly)
- RFC 2869 (partly)

DHCPv4

- RFC 951 (partly)
- RFC 1542 (partly)
- RFC 2131 (partly)

- RFC 2132 (partly)
- RFC 3046 (partly)

DHCPv6

- RFC 8415 (partly)

Access Line Information

The access line identification and characterization information are defined in the Broadband Forum (BBF) formerly known DSL Forum attributes including Agent-Remote-Id and Agent-Circuit-Id.

See the following references for more information about access line attributes.

- RFC 4679 DSL Forum Vendor-Specific RADIUS Attributes
- RFC 6320 ANCP (partly)
- Broadband Forum TR-101 (partly)
- draft-lihawi-ancp-protocol-access-extension-04 (partly)

L2TPv2

ℹ RFC and draft compliance are partial except as specified.

RFC 2661 - Layer Two Tunneling Protocol (L2TPv2)

RFC compliant L2TPv2 Access Concentrator (LAC) with the following protocol limitations:

- No support for LNS initiated outbound calls (OCRQ, OCRP and OCCN)
- No support for WAN-Error-Notify (WEN) Messages send by LAC to LNS
- No support for Set-Link-Info (SLI) Messages send by LNS to LAC
- No support for L2TP over IPv6
- No support for L2TP offset values other than 0.

RFC 5515 - L2TP Access Line Information AVP Extensions

- Support for access line AVP send (LAC) and received (LNS) as part of the L2TP

Incoming-Call- Request (ICRQ) message.

- Response to Connect-Speed-Update-Request (CSURQ) L2TP messages is currently not supported.

RFC 2868 - RADIUS Attributes for Tunnel Protocol Support

RADIUS support for L2TP with the following limitations:

- No support of FQDN format for IP addresses
- No support Tunnel-Medium-Type other than IPv4

RFC 3145 - L2TP Disconnect Cause Information

Send meaningful disconnect cause information to LNS and display received disconnect cause information for tunnels and sessions.

Supported Hardware

You can find more detailed information about what RtBrick features are supported on each hardware platform, see the *Platform Guide*.

4.2. RBFS RADIUS Services

4.2.1. RADIUS Services Overview

The modular, scalable subscriber management that RtBrick calls the next generation access infrastructure (ng-access) provides support for protocols such as PPPoE, IPoE, L2TP and RADIUS.

The subscriber management infrastructure provides the next generation of internet access protocols designed for carrier grade services in regards to scalability and robustness. One of the challenges for carrier networks is interwork with numerous client devices which requires a well implemented, industry proven access protocol stack, including support for all relevant RFCs.

This implementation is designed to be a set of distributed services for increased scaling and stability. The subscriber management implementation has three components:

- subscriberd: subscriber management and AAA (local, RADIUS, and other support)

- pppoed: PPPoE/PPP session handling
- l2tpd: L2TP tunnel and session handling

The subscriber daemon (subscriberd) is the central application, keeping the current subscriber state as well as being responsible for Authentication, Authorization and Accounting (AAA).

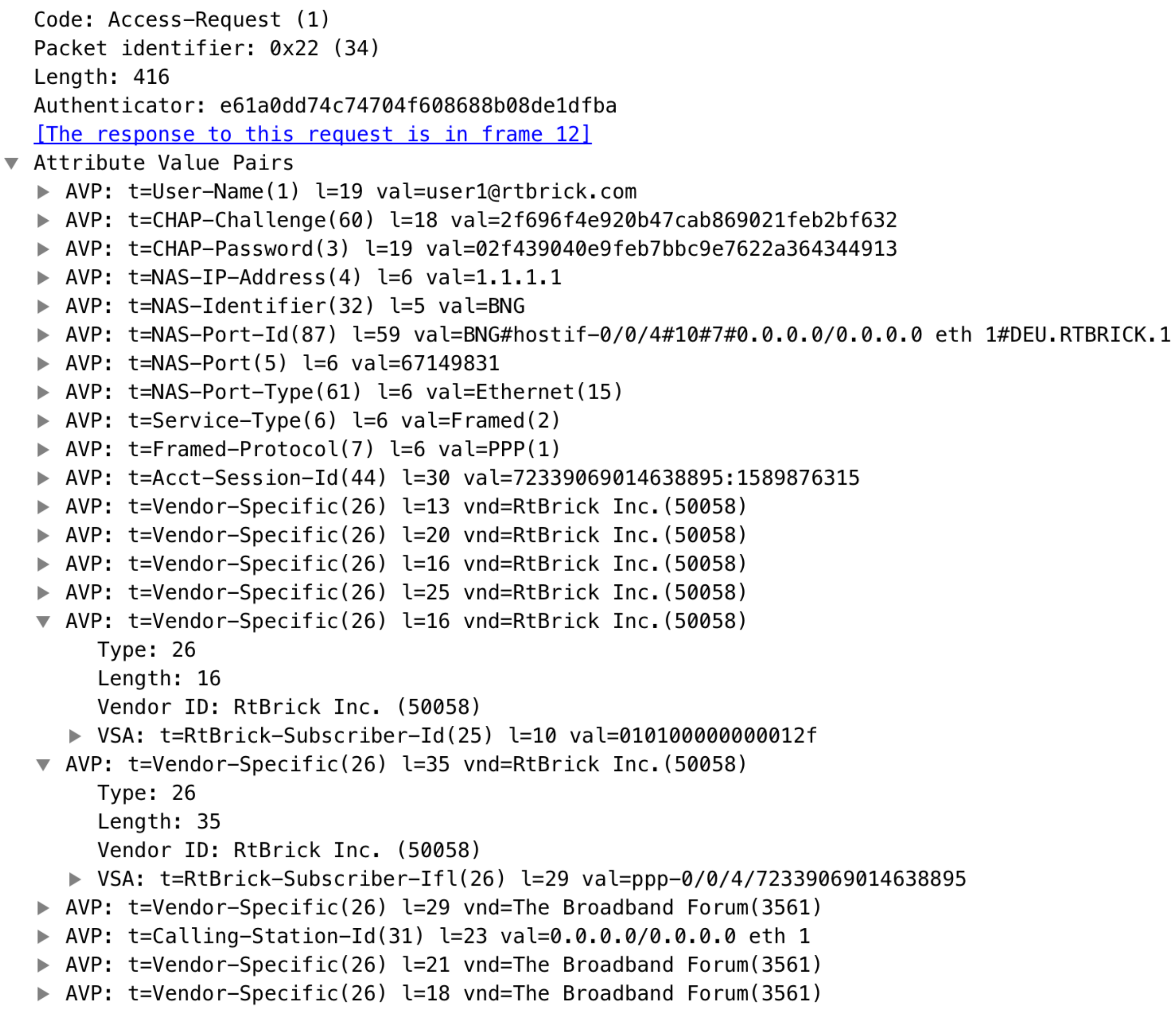
This document describes the subscriber management RADIUS service implementation on RBFS. The term subscriber describes an access user or session from a higher level decoupled from underlying protocols like PPPoE or IPoE. Subscribers in RBFS can be managed locally or via RADIUS where this document considers RADIUS only. Each subscriber is uniquely identified by a 64bit number called subscriber-id. Remote Authentication Dial-In User Service (RADIUS) is a networking protocol that provides centralized Authentication, Authorization and Accounting (AAA) management for all types of subscribers (PPPoE, or IPoE). RADIUS servers can perform as authentication and accounting servers or change of authorization (CoA) clients. Authentication servers maintain authentication records for subscribers. The subscriber daemon requests authentication in RADIUS access-request messages before permitting subscribers access. Accounting servers handle accounting records for subscribers. The subscriber daemon transmits RADIUS accounting-start, interim and stop messages to the servers. Accounting is the process of tracking subscriber activity and network resource usage in a subscriber session. This includes the session time called time accounting and the number of packets and bytes transmitted during the session called volume accounting.

A RADIUS server can behave as a change of authorization (CoA) client allowing dynamic changes for subscriber sessions. The subscriber daemon supports both RADIUS CoA messages and disconnect messages. CoA messages can modify the characteristics of existing subscriber sessions without loss of service, disconnect messages can terminate subscriber sessions.

Each RADIUS request from subscriber daemon includes the RADIUS accounting-session-id attribute (type 44) with a format which is configurable in the AAA profile and includes at least the subscriber-id to identify the corresponding subscriber. The default format (<subscriber-id>.<timestamp>) includes also an Unix timestamp to ensure that the tuple of NAS-Identifier (e.g. hostname) and Accounting-Session-Id is global unique to be usable as key in RADIUS databases.

Additionally to subscriber-id and accounting-session-id each subscriber consists also of a subscriber-ifl build based on physical port information and subscriber-id (ifp: ifp-0/0/1 and subscriber-id: 72339069014638610 subscriber-ifl: ppp-0/0/1/72339069014638610) which is required as handle in the RBFS forwarding infrastructure. All those three informations are part of the RADIUS access-request message:

- accounting-session-id: standard attribute 44
- subscriber-id: RtBrick VSA (26-50058-25 RtBrick-Subscriber-Id)
- subscriber-ifl: RtBrick VSA (26-50058-26 RtBrick-Subscriber-Ifl)



```
    Code: Access-Request (1)
    Packet identifier: 0x22 (34)
    Length: 416
    Authenticator: e61a0dd74c74704f608688b08de1dfba
    [The response to this request is in frame 12]
  ▼ Attribute Value Pairs
    ▶ AVP: t=User-Name(1) l=19 val=user1@rtbrick.com
    ▶ AVP: t=CHAP-Challenge(60) l=18 val=2f696f4e920b47cab869021feb2bf632
    ▶ AVP: t=CHAP-Password(3) l=19 val=02f439040e9feb7bbc9e7622a364344913
    ▶ AVP: t=NAS-IP-Address(4) l=6 val=1.1.1.1
    ▶ AVP: t=NAS-Identifier(32) l=5 val=BNG
    ▶ AVP: t=NAS-Port-Id(87) l=59 val=BNG#hostif-0/0/4#10#7#0.0.0.0/0.0.0.0 eth 1#DEU.RTBRICK.1
    ▶ AVP: t=NAS-Port(5) l=6 val=67149831
    ▶ AVP: t=NAS-Port-Type(61) l=6 val=Ethernet(15)
    ▶ AVP: t=Service-Type(6) l=6 val=Framed(2)
    ▶ AVP: t=Framed-Protocol(7) l=6 val=PPP(1)
    ▶ AVP: t=Acct-Session-Id(44) l=30 val=72339069014638895:1589876315
    ▶ AVP: t=Vendor-Specific(26) l=13 vnd=RtBrick Inc.(50058)
    ▶ AVP: t=Vendor-Specific(26) l=20 vnd=RtBrick Inc.(50058)
    ▶ AVP: t=Vendor-Specific(26) l=16 vnd=RtBrick Inc.(50058)
    ▶ AVP: t=Vendor-Specific(26) l=25 vnd=RtBrick Inc.(50058)
    ▼ AVP: t=Vendor-Specific(26) l=16 vnd=RtBrick Inc.(50058)
        Type: 26
        Length: 16
        Vendor ID: RtBrick Inc. (50058)
      ▶ VSA: t=RtBrick-Subscriber-Id(25) l=10 val=010100000000012f
    ▼ AVP: t=Vendor-Specific(26) l=35 vnd=RtBrick Inc.(50058)
        Type: 26
        Length: 35
        Vendor ID: RtBrick Inc. (50058)
      ▶ VSA: t=RtBrick-Subscriber-Ifl(26) l=29 val=ppp-0/0/4/72339069014638895
    ▶ AVP: t=Vendor-Specific(26) l=29 vnd=The Broadband Forum(3561)
    ▶ AVP: t=Calling-Station-Id(31) l=23 val=0.0.0.0/0.0.0.0 eth 1
    ▶ AVP: t=Vendor-Specific(26) l=21 vnd=The Broadband Forum(3561)
    ▶ AVP: t=Vendor-Specific(26) l=18 vnd=The Broadband Forum(3561)
```

The subscriber-id is an unsigned 64bit integer which is shown as a hex number in Wireshark.

Each subscriber is formed based on configuration profiles and individual settings retrieved via RADIUS. Conflicts between RADIUS defined attributes and profile attributes are solved by prioritizing those received from RADIUS which is common best practices for broadband access concentrators. New subscribers are signalled via RADIUS access-request and either accepted by RADIUS access-accept or rejected by RADIUS access-reject message from RADIUS server. The RADIUS access-accept includes all attributes required to form the subscriber like IP addresses, DNS servers and referenced configuration profiles. Some of those attributes can be changed by RADIUS dynamically using CoA requests without disconnecting the subscriber.

Some of those attributes refer to RADIUS services which are described in detail in this document. The term RADIUS services described the ability to dynamically control the properties of a session via RADIUS used by access providers to build their different products and services based on their network infrastructure. This is used to control QoS settings like shaper rates dynamically based on changed line quality or possible volume quotas. It is also used to dynamically enable or disable services like Voice or IPTV on a per subscriber basis.

The RADIUS accounting accuracy should comply with §45g of the German Telecommunications Act (TKG) further named by TKG§45. This applies for time and volume based products and services. Products or services with unlimited volumes but with changed properties after a certain amount of traffic are considered as volume based products as well. One example here is a fair use policy which enforces a rate limit after a certain volume. The requirements for time based accounting apply only to products or services which are charged by time or changed properties after a certain time which is not very common in the market today.

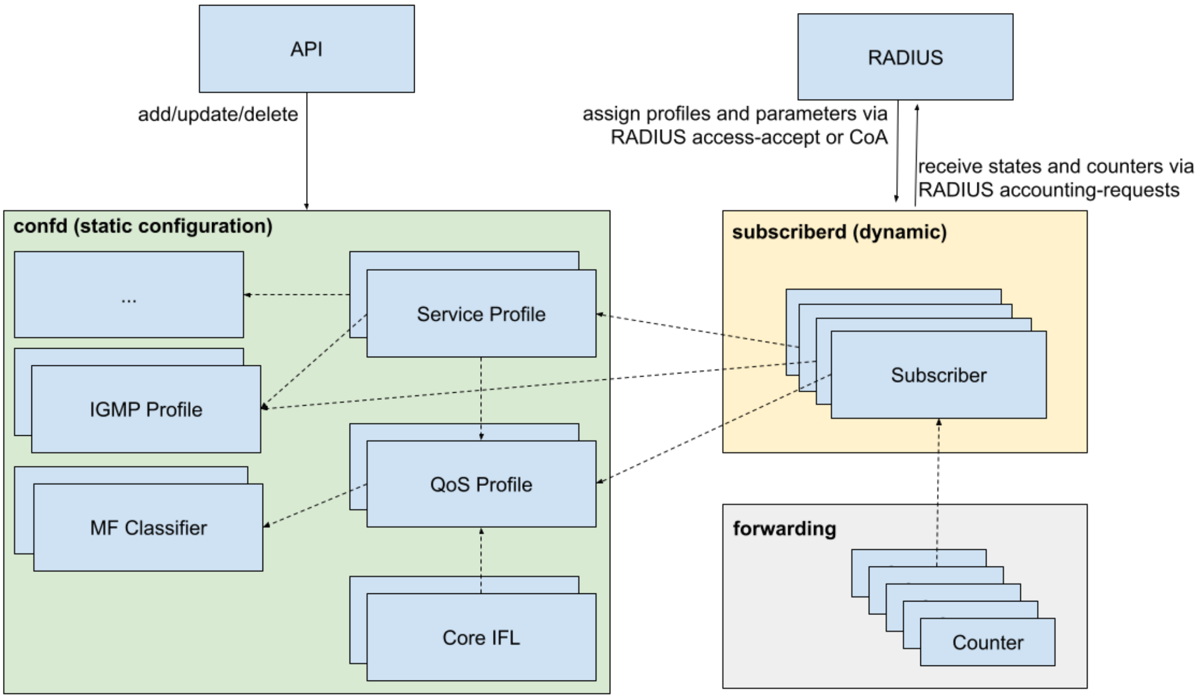
Supported Platforms

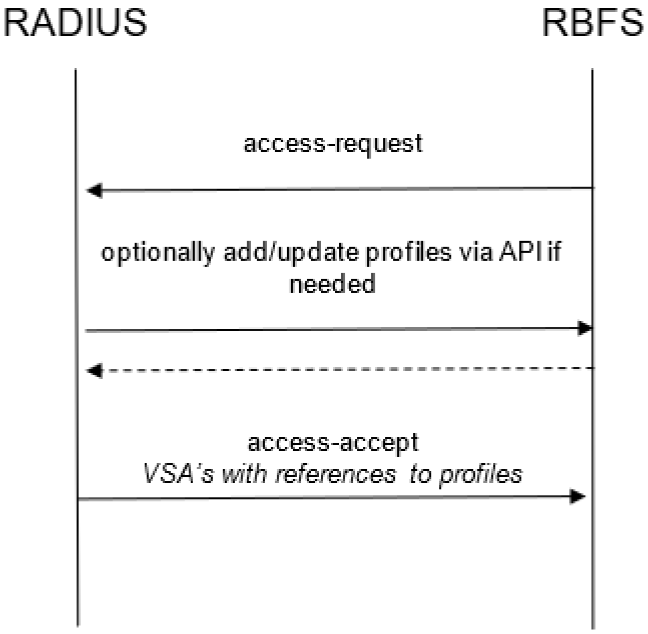
Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

4.2.2. RADIUS Service Solution Overview

The general concept of RADIUS services in RtBrick FullStack (RBFS) consists of pre-configured profiles in the global static configuration assigned to dynamic subscribers via RADIUS VSA in access-accept and CoA request messages. The profiles are managed as any other configuration in RBFS via corresponding API which is not explained here in detail.

The following pictures show the concept of global static configurations referenced by RADIUS controlled dynamic subscribers.


API
add/update/delete
RADIUS
assign profiles and parameters via RADIUS access-accept or CoA
receive states and counters via RADIUS accounting-requests
confd (static configuration)
...
Service Profile
IGMP Profile
QoS Profile
MF Classifier
Core IFL
subscriberd (dynamic)
Subscriber
forwarding
Counter


RADIUS
RBFS
access-request
optionally add/update profiles via API if needed
access-accept
VSA's with references to profiles

The required profiles and filters should be present before service becomes active which can be done immediately after start-up or between access-request and access-accept as shown in the flow diagram on the left.

4.2.3. RADIUS Control

This chapter explains the subscriber and service-related RBFS extensions and RtBrick VSA related to RADIUS services.

RADIUS Session and Idle Timeout

The session and idle timeout values are initialized with zero, which means infinity or disabled. Those values will be optionally overwritten with the values in AAA configuration profile and RADIUS access-accept if present. The priority is RADIUS attributes over AAA configuration profile.

The session (attribute 27) and idle (attribute 28) timeout RADIUS attributes are defined in RFC 2865.

> ℹ RFC and draft compliance are partial except as specified.

The idle timeout is based on outgoing logical interface statistics (OutLIF) for the subscriber-ifl to determine subscriber inactivity.

RADIUS Routing Instance

VSA 26-50058-137 - RtBrick-Instance

This attribute allows to change the routing instance via RADIUS access-accept.

RADIUS DNS Server

VSA 26-50058-131 - RtBrick-DNS-Primary

VSA 26-50058-132 - RtBrick-DNS-Secondary

VSA 26-50058-134 - RtBrick-DNS-Primary-IPv6

VSA 26-50058-135 - RtBrick-DNS-Secondary-IPv6

Those attributes allow to set the primary and secondary IPv4 and IPv6 DNS servers via RADIUS access-accept. DNS servers already set via access configuration profile will be overwritten by RADIUS.

RADIUS Service Profile

VSA 26-50058-70 - RtBrick-Service-Profile

Subscribers can be associated with a service-profile which defines the actual service properties like QoS or IGMP profiles and configuration settings. The service profile can be assigned or changed using the RtBrick-Service-Profile VSA.

```
RtBrick-Service-Profile = <service-profile>
```

This attribute is supported via RADIUS access-accept and CoA requests.

All dynamic QoS settings like shaper and policer rates will be reset if the new service-profile includes a qos-profile attribute also if active qos-profile and old qos-profile is equal. All QoS settings remain unchanged if the referenced service profile does not include the qos-profile attribute. If the referenced service profile updates the qos-profile attribute and additional shaper or policer rates are included in the same CoA request which updates the service-profile, those shaper and policer settings will be applied to the new QoS configuration profile after reset. This means that we reset all incremental changes done before. In example if Voice shaper rate has changed to another value, after profile change the default value from profile is used.

All dynamic multicast settings like IGMP status, IGMP profile, IGMP version and IGMP max- members will be reset if this attribute is received via CoA. Therefore assigning a new service profile via CoA without IGMP enabled in the service profile will disable IGMP also without sending RtBrick-IGMP-Status.

RADIUS CoA requests referencing a service profile which is not found on device will be rejected with CoA-NAK (invalid-request) but without changing any subscriber QoS settings. This behavior is different for RADIUS access-accept where service profiles are just ignored if not found. In both cases a warning is generated in subscriber daemon log if a referenced service profile is not found.

RADIUS AAA Profile

VSA 26-50058-69 - RtBrick-AAA-Profile

This attribute allows to change the associated AAA profile from RADIUS access-accept. This is primarily used to change the accounting adjustment values which are defined in this profile. Simple example here is to use different adjustment values for L2TP and local terminated PPPoE sessions. Another valid use case is to assign different RADIUS accounting servers for in example L2TP sessions or wholesale customers.

```
RtBrick-AAA-Profile = <aaa-profile>
```

This attribute is supported via RADIUS access-accept only.

RADIUS QoS Profile

VSA 26-50058-62 - RtBrick-QoS-Profile

```
RtBrick-QoS-Profile = <qos-profile>
```

This attribute is supported by RADIUS access-accept and CoA request. The QoS configuration profile can be either selected via service-profile or directly using this attribute which has priority of the service-profile.

All dynamic QoS settings like MFC, queue sizes, shaper and policer rates will be reset if this attribute is present in a CoA request also if new qos-profile and old qos-profile is equal. If additional shaper or policer rates are included in the same CoA request which updates the service-profile, those shaper and policer settings will be applied to the new QoS configuration after reset.

The subscriber management infrastructure does not check if a referenced QoS profile exists or not. This is handled by forwarding infrastructure which continues processing the subscriber QoS settings as soon as the profile was added. In the meantime there is no QoS configuration applied.

RADIUS QoS Parent Scheduler

VSA 26-50058-64 - RtBrick-QoS-Parent-Scheduler

The parent scheduler element of the scheduler-map assigned to the subscriber can be selected with this attribute. If not present, the scheduler-map will be directly bound to the local IFP where the session is established.

This attribute is supported in RADIUS access-accept only (no CoA support) and will set the parent scheduler element of the subscriber.

⚠ Providing a QoS parent scheduler which is not present on the

> corresponding IFP will lead to black howling of all egress data traffic. Control traffic is not impacted and therefore the session will remain.

RADIUS QoS Shaper

VSA 26-50058-63 - RtBrick-QoS-Shaper

Subscribers can be associated with a QoS profile which is assigned via service-profile or directly via corresponding VSA (RtBrick-QoS-Profile). This profile is used to instantiate the QoS resources for the subscriber including schedulers, queues and shapers. The assigned shaper instances can be updated using the RtBrick-QoS-Shaper VSA (attribute 26-50058-63) which will apply to the QoS instance of the corresponding subscriber only, but not to the other subscribers using the same QoS profile. It is only possible to update existing shapers dynamically but it is not possible to create a new shaper via RADIUS.

The RtBrick-QoS-Shaper value is a string which contains a list of multiple shaper settings separated by semicolon. Each shaper setting contains a shaper name, high flow rate and low flow rate separated by comma. The actual shaper rate is the sum of high and low flow rate.

```
RtBrick-QoS-Shaper = <shaper-name>,<high-kbps>,<low-kbps>;<shaper-name>,...
```

This attribute can be also used as a key-value list which is automatically recognized by RBFS.

```
RtBrick-QoS-Shaper = name=<shaper-name>,high=<high-kbps>,low=<low-kbps>;...
```

This attribute is supported by RADIUS access-accept and CoA request.

Updating a single shaper (e.g. voice_shaper) via CoA does not require to include other shapers meaning that only the shapers included in the attribute will be updated and all other shapers remain unchanged.

Assume the following example which adapts the session and voice shaper instance of a subscriber.

```
supervisor@rtbrick: op> show config forwarding-options class-of-service shaper
shaper_session
```

```
{
  "rtbrick-config:shaper": {
    "shaper-name": "shaper_session",
    "shaping-rate-high": 48000,
    "shaping-rate-low": 2000
  }
}
supervisor@rtbrick: op> show config forwarding-options class-of-service shaper
shaper_voice
{
  "rtbrick-config:shaper": {
    "shaper-name": "shaper_voice",
    "shaping-rate-high": 1000,
    "shaping-rate-low": 0
  }
}
```

RADIUS VSA Change Session Shaper Only

```
RtBrick-QoS-Shaper: shaper_session,14000,2000
```

RADIUS VSA Change Session and Voice Shaper

```
RtBrick-QoS-Shaper: shaper_session,14000,2000;shaper_voice,2000
```

All active dynamic shapers are stored in the table **global.access.1.subscriber.qos.shaper** to handover those information to forwarding infrastructure. This table can be used to verify the dynamic shapers which are active for a given subscriber.

The CLI command show subscriber <id> qos displays those information in a nice human readable format.

```
supervisor@rtbrick: op> show subscriber 72339069014638610 qos
Subscriber-Id: 72339069014638610
    Interface: ifp-0/0/1
    Outer VLAN: 128
    Inner VLAN: 7
    IFL: ppp-0/0/1/72339069014638610
    Profile: qos_profile
    Parent: pon1
    Dynamic Shaper: shaper_voice
        Rate Low: 0 kbps
        Rate High: 2000 kbps
    Dynamic Shaper: shaper_session
        Rate Low: 2000 kbps
        Rate High: 14000 kbps
```

⚠ A shaper rate of 0 means infinity!

RADIUS QoS Policer

VSA 26-50058-65 - RtBrick-QoS-Policer

The RtBrick-QoS-Policer value is a string which contains a list of multiple policer level settings separated by semicolon. Each setting contains a level, cir, cbs, pir, pbs, max-cir and max-pir separated by comma.

RtBrick-QoS-Policer = <level>,<cir>,<cbs>,<pir>,<pbs>,<max-cir>,<max-pir>;<level>…

Example:
RtBrick-QoS-Policer = 1,2000,200;2,8000,800;3,0,800;4,0,800

- **level**: 1 (highest priority) to 4 (lowest priority)
- **cir**: Ingress policer committed information rate (kbps)
- **cbs**: Ingress policer committed burst size (kbits)
- **pir**: Ingress policer peak information rate (kbps)
- **pbs**: Ingress policer peak burst size (kbits)
- **max-cir**: max ingress policer committed information rate (kbps)
- **max-pir**: max ingress policer peak information rate (kbps)

If PIR and PBS are not defined, the values from CIR and CBS are used as PIR and PBS as well. The max CIR and max PIR attributes are optional default set to unlimited.

This attribute can be also used as a key-value list which is automatically recognized by RBFS.

RtBrick-QoS-Policer = level=<level>,cir=<cir>,cbs=<cbs>,pir=<pir>,pbs=<pbs>,max-cir=<max-cir>,max-pir=<max-pir>;…

Example:
RtBrick-QoS-Policer = level=4,cir=1m,cbs=256;cir=2m,cbs=512,level=3

This attribute is supported by RADIUS access-accept and CoA request.

Updating a single policer level via CoA does not require to include other levels meaning that only the levels included in the attribute will be updated and all other levels remain unchanged. It is only possible to update existing policer levels dynamically but it is not possible to create a new level via RADIUS.

Assume the following example which adapts the just one level as well as all levels of a subscriber.

```
supervisor@rtbrick: op> show config forwarding-options class-of-service policer
policer-residential
{
  "rtbrick-config:policer": {
    "policer-name": "policer-residential",
    "level1-rates": {
      "cir": 1000,
      "cbs": 100,
      "pir": 1000,
      "pbs": 100
    },
    "level2-rates": {
      "cir": 4000,
      "cbs": 400,
      "pir": 4000,
      "pbs": 400
    },
    "level3-rates": {
      "cir": 0,
      "cbs": 800,
      "pir": 0,
      "pbs": 800
    },
    "level4-rates": {
      "cir": 0,
      "cbs": 800,
      "pir": 0,
      "pbs": 800
    },
    "levels": 4,
    "type": "two-rate-three-color"
  }
}
```

RADIUS VSA Change Level 2 Only

```
RtBrick-QoS-Policer: 2,12000,1200
```

RADIUS VSA Change all Levels

```
RtBrick-QoS-Policer: 1,2000,200;2,8000,800;3,0,800;4,0,800
```

All active dynamic policer level settings are stored in the table **global.access.1.subscriber.qos** to handover those information to forwarding infrastructure. This table can be also used to verify the dynamic policer settings for a given subscriber.

The CLI command show subscriber <id> qos displays those information in a nice human readable format.

```
supervisor@rtbrick: op> show subscriber 72339069014638610 qos
Subscriber-Id: 72339069014638610
    Interface: ifp-0/0/1
    Outer VLAN: 128
    Inner VLAN: 7
    IFL: ppp-0/0/1/72339069014638610
    Profile: qos_profile
    Parent: pon1
    Dynamic Ingress Policer Level 1:
        CIR: 2000 kbps CBS: 200 kbps
        PIR: 2000 kbps PBS: 200 kbps
    Dynamic Ingress Policer Level 2:
        CIR: 8000 kbps CBS: 800 kbps
        PIR: 8000 kbps PBS: 800 kbps
    Dynamic Ingress Policer Level 3:
        CIR: 0 kbps CBS: 800 kbps
        PIR: 0 kbps PBS: 800 kbps
    Dynamic Ingress Policer Level 4:
        CIR: 0 kbps CBS: 800 kbps
        PIR: 0 kbps PBS: 800 kbps
    Dynamic Shaper: shaper_voice
        Rate Low: 0 kbps
        Rate High: 2000 kbps
    Dynamic Shaper: shaper_session
        Rate Low: 2000 kbps
        Rate High: 14000 kbps
```

RADIUS QoS MFC

VSA 26-50058-66 - RtBrick-QoS-MFC

The multifield classifier can be either derived from qos-profile or directly using this attribute which has priority of the qos-profile.

RtBrick-QoS-MFC = <mfc-name>

This attribute is supported by RADIUS access-accept and CoA request.

The subscriber management infrastructure does not check if a referenced multifield classifier exists or not. This is handled by forwarding infrastructure which continues processing the subscriber QoS settings as soon as the classifier was added.

The string received in these attributes should be stored as mf_classifier_name in the RADIUS attribute object and as well as in the corresponding subscriber-qos object.

RADIUS QoS Queues

VSA 26-50058-67 - RtBrick-QoS-Queues

Subscribers can be associated with a QoS profile which is assigned via service-profile or directly via corresponding VSA (RtBrick-QoS-Profile). This profile is used to instantiate the QoS resources for the subscriber including schedulers, queues and shapers. The assigned queue instances can be updated using the RtBrick-QoS-Queues VSA (attribute 26-50058-67) which will apply to the QoS instance of the corresponding subscriber only, but not to the other subscribers using the same QoS profile. It is only possible to update existing queues dynamically but it is not possible to add queues via RADIUS.

The RtBrick-QoS-Queues value is a string which contains a list of multiple queue settings separated by semicolon. Each queue setting contains a queue name and queue size in bytes separated by comma.

```
RtBrick-QoS-Queues = <queue-name>,<size-bytes>;<queue-name>,<size-bytes>;...
```

This attribute can be also used as a key-value list which is automatically recognized by RBFS.

```
RtBrick-QoS-Queues = name=<queue-name>,size=<size-bytes>;name=...
```

This attribute is supported by RADIUS access-accept and CoA request.

Updating a single queue (e.g. best-effort) via CoA does not require to include other queues meaning that only the queues included in the attribute will be updated and all other queues remain unchanged.

The subscriber management infrastructure does not check if a referenced queue exists or not. This is handled by forwarding infrastructure which continues processing the subscriber QoS settings as soon as the queue was added.

All active dynamic queues are stored in the table **global.access.1.subscriber.qos.queue** to handover those information to forwarding infrastructure. This table can be also used to verify the dynamic queues which are active for a given subscriber.

The CLI command show subscriber <id> qos displays those information in a nice human readable format.

```
supervisor@rtbrick: op> show subscriber 72339069014638610 qos
Subscriber-Id: 72339069014638610
    Interface: ifp-0/0/1
    Outer VLAN: 128
    Inner VLAN: 7
    IFL: ppp-0/0/1/72339069014638610
    Profile: qos_profile
    Parent: pon1
    Dynamic Ingress Policer Level 1:
        CIR: 2000 kbps CBS: 200 kbps
        PIR: 2000 kbps PBS: 200 kbps
    Dynamic Ingress Policer Level 2:
        CIR: 8000 kbps CBS: 800 kbps
        PIR: 8000 kbps PBS: 800 kbps
    Dynamic Ingress Policer Level 3:
        CIR: 0 kbps CBS: 800 kbps
        PIR: 0 kbps PBS: 800 kbps
    Dynamic Ingress Policer Level 4:
        CIR: 0 kbps CBS: 800 kbps
        PIR: 0 kbps PBS: 800 kbps
    Dynamic Shaper: shaper_voice
        Rate Low: 0 kbps
        Rate High: 2000 kbps
    Dynamic Shaper: shaper_session
        Rate Low: 2000 kbps
        Rate High: 14000 kbps
    Dynamic Queue: voice
        Size: 200000 byte
```

RADIUS IGMP Attributes

IGMP service can be dynamically enabled on a subscriber using Radius IGMP Attributes. These attributes are supported by RADIUS access-accept and CoA requests. Changes via CoA will reset the existing IGMP configuration therefore changing one attribute like max members requires to send all other attributes as well.

Following IGMP related attributes are supported:

VSA 26-50058-71 - RtBrick-IGMP-Status

This attribute can dynamically enable/disable IGMP for a subscriber

Value	Code	Description
DISABLED	0	Disable IGMP
ENABLED	1	Enable IGMP

VSA 26-50058-72 - RtBrick-IGMP-Profile

```
RtBrick-IGMP-Profile = <igmp-profile>
```

This attribute specifies the IGMP-profile to be associated with the subscriber. IGMP profile is configured locally in IGMP with all the IGMP related attributes. This attribute can dynamically associate such a locally configured profile to a subscriber

The subscriber management infrastructure does not check if a referenced IGMP profile exists or not. This is handled by IGMP infrastructure which continues processing the subscriber IGMP settings as soon as the profile was added.

VSA 26-50058-73 - RtBrick-IGMP-Version

This attribute can specify the version of IGMP for a subscriber.

Value	Code	Description
V1	1	IGMP Version 1 (not supported)
V2	2	IGMP Version 2
V3	3	IGMP Version 3 (Default version if this attribute is not set)

VSA 26-50058-74 - RtBrick-IGMP-Max-Members

This integer attribute can specify the number of parallel streaming (maximum IGMP membership) for a subscriber.

All active dynamic IGMP attributes are stored in the table **global.access.1.subscriber.igmp.service** to handover those information to IGMP. This table can be also used to verify the dynamic IGMP attributes which are active for a given subscriber.

The CLI command show subscriber <id> igmp displays those information in a nice human readable format.

```
supervisor@rtbrick: op> show subscriber 72339069014638706 igmp
Subscriber-Id: 72339069014638706
    Interface: ifp-0/0/1
    Outer VLAN: 128
    Inner VLAN: 7
    IFL: ppp-0/0/3/72339069014638706
    Version: IGMPv3
    Profile: iptv-basic
    Max Members: 6
```

RADIUS Connection Status Message Attribute

VSA 26-50058-139 - RtBrick-Connection-Status-Message

This attribute allows to send a connection status message string via PPP LCP vendor extension (RFC2513) to the client.

The connection status message is typically used to inform clients about the available bandwidth via RADIUS access-accept or CoA request.

RtBrick-Connection-Status-Message = <string>

Adding or changing the connection status message triggers to send a LCP vendor specific packet with OUI set to f4:1e:5e (RtBrick Inc.) and kind 0x01.

RFC 2513 - Vendor Specific Packet

```
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |     Code      |  Identifier   |            Length             |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                         Magic-Number                          |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                      OUI                      |     Kind      |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |    Value(s) ...
 +-+-+-+-+-+-+-+-+
```

The value is encoded as TLV with type set to 0x01 and length including type and length field.

The actual status message string is limited to 247 bytes.

Connection-Status-Message Option

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     | Connection-Status-Message     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| ...
+-+-+
```

PPP clients with support for connection status message will respond with a vendor specific packet of kind 0x02 and the same OUI but no value to acknowledge the packet. All other clients either ignore or reject the message using PPP LCP code reject.

The connection status message will be retried 10 times with an interval of 3 seconds until the client has acknowledged or rejected the packet.

The status of the connection status message can be verified with the following command.

```
supervisor@leaf1: op> show pppoe session 72339069014638706 detail
Subscriber-Id: 72339069014638706
    State: ESTABLISHED
    Uptime: Mon Sep 13 09:29:42 GMT +0000 2021 (0:14:57.625027)
    Interface: ifp-0/0/1
    Outer VLAN: 128
    Inner VLAN: 7
...
    PPP LCP:
        State: OPENED
        Negotiated Protocols: PAP, IPCP, IP6CP
        Negotiated Parameters: MRU, AUTH, MAGIC
        Magic Number: 130827225 Peer: 2835676915
        MRU: 1492 Peer: 1492
        Echo Interval: 30 seconds
    PPP LCP Connection Status Message:
        State: ACCEPTED
        Message: SRU=10000#SRD=100000#
...
```

RADIUS Ascend-Data-Filter Attribute

VSA 26-529-242 - Ascend-Data-Filter

The Ascend-Data-Filter attribute describes per subscriber filter terms in a simple binary format as described in the following table. Multiple of those attributes in a single RADIUS access-accept or CoA request message are combined to a dynamic filter where each attribute itself is one filter term. The order of those attributes in the RADIUS message is used to order the corresponding terms in the filter. This means that the first filter in the RADIUS packet has priority over the next and or last filter in the RADIUS packet.

Changes in such a filter via CoA requires that all attributes of the new filter must be sent. Therefore adding a single filter term requires sending the existing filter terms plus the new one or general speaking sending the intended target filter.

This filter is installed as a global packet filter placed before policer in ingress and before scheduler in egress to prevent that traffic dropped here is counted in accounting or consuming rate credits.

Field	Bytes	Values	Comments
Type	1	0 = ignore 1 = IPv4 3 = IPv6	
Action	1	0 = discard 1 = accept 0x20 = redirect	
Direction	1	0 = egress 1 = ingress	
Reserved	1	0	

Source
Address
	IPv4 = 4
	IPv6 = 16
		source address	Match on source address is not supported for filters in ingress direction and automatically replaced with the subscriber addresses.

Field	Bytes	Values	Comments

Destination
Address
	IPv4 = 4
	IPv6 = 16
		destination address	Match on destination address is not supported for filters in egress direction and automatically replaced with the subscriber addresses.

Source Prefix Length	1	0 = ignore prefix length	The number of leading one bits in the source address mask. Specifies the bits of interest.
Destination Prefix Length	1	0 = ignore prefix length	The number of leading one bits in the destination address mask. Specifies the bits of interest.
Protocol	1	0 = ignore IPv4 protocol number IPv6 next header	
Reserved	1	0	
Source Port	2	port number	UDP/TCP source port in network byte order (big endian)
Destination Port	2	port number	UDP/TCP destination port in network byte order (big endian)

Field	Bytes	Values	Comments
Source Port Qualifier	1	0 = no compare 1 = less than 2 = equal to 3 = greater than 4 = not equal to	The options 1, 3 and 4 are not implemented and mapped to 0 if received.
Destination Port Qualifier	1	0 = no compare 1 = less than 2 = equal to 3 = greater than 4 = not equal to	The options 1, 3 and 4 are not implemented and mapped to 0 if received.
Not Used	0 - N	0	Trailing bytes after destination port qualifier ignored.

The filter can be deleted dynamically by sending a single Ascend-Data-Filter attribute with a zero value.

Example:

```
Ascend-Data-Filter =
0x01000100000000000a0afffe002000000000000000000000000000000000

    01          IPv4            // type
    00          discard         // action
    01          ingress         // direction
    00          -               // reserved
    00000000    -               // source address
    0a0afffe    198.51.100.101  // destination address
    00          -               // source prefix length
    20          /32             // destination prefix length
    00          -               // protocol
    00          -               // reserved
    0000        -               // source port
    0000        -               // destination port
    00          -               // source port qualifier
    00          -               // destination port qualifier
    00000...    -               // ignored
```

For ingress filters, it is not permitted to match based on the source address which is automatically replaced with the subscriber addresses. A similar limitation is valid for egress filters matching on destination address which is also replaced with subscriber addresses.

A new proprietary filter action 0x20 (32) is added for the redirect service. This can be used to enable HTTP Redirect for a subscriber using the ADF.

All active subscriber filters are stored in the table **global.access.1.subscriber.filter** to hand over that information to the forwarding infrastructure. This table can be also used to verify the filters which are active for a given subscriber.

The CLI command show subscriber <id> acl displays those filters in a nice human-readable format.

RADIUS Access Line Attributes

The access line identification and characterization information are defined in the Broadband Forum (BBF) formerly known DSL Forum attributes including Agent-Remote-Id and Agent-Circuit-Id.

See the following references for more information about access line attributes.

- RFC 4679 DSL Forum Vendor-Specific RADIUS Attributes
- RFC 6320 ANCP
- Broadband Forum TR-101
- draft-lihawi-ancp-protocol-access-extension-04

Those attributes will be sent in RADIUS access and accounting requests to the RADIUS server if learned from the underlying access protocol like PPPoE (BBF TR-101).

RBFS also provides the possibility to set those values via RADIUS by using the same attributes in the RADIUS access-accept or CoA request message.

Access Line Attributes Supported (RADIUS, PPPoE and L2TP):

Attribute	Name
26-3561-1	Agent-Circuit-Id
26-3561-2	Agent-Remote-Id
26-3561-3	Access-Aggregation-Circuit-ID-ASCII
26-3561-6	Access-Aggregation-Circuit-ID-ASCII
26-3561-18	PON-Access-Line-Attributes
26-3561-129	Actual-Data-Rate-Upstream
26-3561-130	Actual-Data-Rate-Downstream
26-3561-131	Minimum-Data-Rate-Upstream
26-3561-132	Minimum-Data-Rate-Downstream
26-3561-133	Attainable-Data-Rate-Upstream
26-3561-134	Attainable-Data-Rate-Downstream
26-3561-135	Maximum-Data-Rate-Upstream
26-3561-136	Maximum-Data-Rate-Downstream
26-3561-137	Minimum-Data-Rate-Upstream-Low-Power
26-3561-138	Minimum-Data-Rate-Downstream-Low-Power
26-3561-139	Maximum-Interleaving-Delay-Upstream
26-3561-140	Actual-Interleaving-Delay-Upstream
26-3561-141	Maximum-Interleaving-Delay-Downstream
26-3561-142	Actual-Interleaving-Delay-Downstream
26-3561-144	Access-Loop-Encapsulation
26-3561-145	DSL-Type
26-3561-151	PON-Access-Type
26-3561-155	Expected-Throughput-Upstream
26-3561-156	Expected-Throughput-Downstream
26-3561-157	Attainable-Expected-Throughput-Upstream

Attribute	Name
26-3561-158	Attainable-Expected-Throughput-Downstream
26-3561-159	Gamma-Data-Rate-Upstream
26-3561-160	Gamma-Data-Rate-Downstream
26-3561-161	Attainable-Gamma-Data-Rate-Upstream
26-3561-161	Attainable-Gamma-Data-Rate-Downstream
26-3561-176	ONT-ONU-Average-Data-Rate-Downstream
26-3561-177	ONT-ONU-Peak-Data-Rate-Downstream
26-3561-178	ONT-ONU-Maximum-Data-Rate-Upstream
26-3561-179	ONT-ONU-Assured-Data-Rate-Upstream
26-3561-180	PON-Tree-Maximum-Data-Rate-Upstream
26-3561-181	PON-Tree-Maximum-Data-Rate-Downstream

This table includes already the new attributes defined in draft-lihawi-ancp-protocol-access-extension-04 which results in possible changes in case the draft is updated.

Changes in at least one of those line attributes via CoA trigger a RADIUS interim accounting request with the new values.

Changes in actual data rate upstream/downstream via CoA request may trigger the L2TP LAC to send a CSUN request with updated connection speed to the LNS depending on the actual L2TP configuration.

RADIUS L2TP

Tunneling of PPPoE sessions via L2TPv2 (L2TP LAC) is supported on RBFS and can be controlled via RADIUS as described in RFC 2868 RADIUS Attributes for Tunnel Protocol Support with some limitations:

- No support of FQDN format for IP addresses
- No support Tunnel-Medium-Type other than IPv4

To establish a PPPoE session via L2TP, the tunnel-type must be configured as L2TP. This configuration can be achieved either for local users or by utilizing the corresponding tunnel-type attribute through RADIUS.

Defining an L2TP configuration profile is essential, which can be referenced through an access configuration profile or by employing the appropriate RADIUS VSA. The actual tunnels may either be defined locally via an L2TP pool configuration or set up dynamically through RADIUS.

```
# FreeRADIUS
# RADIUS authenticated with local defined tunnels.
"radius@l2tp" Cleartext-Password := "test"
    Service-Type = Framed-User,
    Framed-Protocol = PPP,
    Tunnel-Type:0 = L2TP

# RADIUS authenticated with dynamic tunnels.
"tunnel@rl2tp" Cleartext-Password := "test"
    Service-Type = Framed-User,
    Framed-Protocol = PPP,
    Tunnel-Type:1 = L2TP,
    Tunnel-Client-Endpoint:1 = "192.0.2.1",
    Tunnel-Server-Endpoint:1 = "192.0.2.2",
    Tunnel-Client-Auth-Id:1 = "bng",
    Tunnel-Server-Auth-Id:1 = "lns",
    Tunnel-Preference:1 = 100,
    Tunnel-Password:1 = "test",
    Tunnel-Type:2 += L2TP,
    Tunnel-Client-Endpoint:2 += "192.0.2.3",
    Tunnel-Server-Endpoint:2 += "192.0.2.4",
    Tunnel-Client-Auth-Id:2 += "BNG",
    Tunnel-Server-Auth-Id:2 += "LNS2",
    Tunnel-Preference:2 += 200,
    Tunnel-Password:2 += "test"
```

In addition to the standard attributes, the following vendor-specific attributes are supported for L2TP.

> ℹ RFC and draft compliance are partial except as specified.

VSA 26-50058-40 - RtBrick-L2TP-Pool

Instead of RADIUS tunnel attributes it is also possible to configure local L2TP tunnel pools and assign them with this attribute.

VSA 26-50058-41 - RtBrick-L2TP-Profile

The default L2TP configuration profile assigned in the access configuration profile can be changed in RADIUS access-accept to allow different L2TP configurations based on tunnel endpoints. This is typically used to enable or disable L2TP CSUN updates (RFC5515).

VSA 26-50058-42 - RtBrick-L2TP-Tx-Connect-Speed

VSA 26-50058-43 - RtBrick-L2TP-Rx-Connect-Speed

Those attributes are supported in RADIUS access-accept and CoA requests to set the corresponding L2TP to connect speed values directly. Per default, the L2TP connect speed is derived from the actual data rate upstream/downstream of the configured source (PPPoE-IA or RADIUS) if present or set to port speed which can be overridden with the connect speed attributes. Changes in connect speed via CoA request will trigger the L2TP LAC to send a CSUN request to the LNS if enabled.

Those two attributes are defined as 4-byte integers with speed defined in kbits.

RADIUS Lawful Interception

Lawful interception (LI) refers to the facilities in telecommunications and telephone networks that allow law enforcement agencies (LEA) with court orders or other legal authorization to selectively intercept individual subscribers. The term mediation device (MED) used in this document describes the element which receives and optionally converts the intercepted traffic into the format expected by the law enforcement agencies of the corresponding countries.

All of the following attributes must be present in RADIUS access-accept or CoA request to control lawful interception (LI) via RADIUS. Those attributes are salt encrypted using the algorithm described in RFC 2868 for the Tunnel Password. This encryption algorithm is defined for RADIUS access-accept messages only. To support CoA requests the request authenticator should be replaced with 16 zero bytes which is common industry standard.

The LI action NOOP can be used to obfuscate lawful interception requests (fake requests) to prevent just the presence of those attributes indicating that a subscriber is intercepted. LI requests via RADIUS will show up in the same table as requests via REST or HTTP RPC API (global.access.1.li_request).

Failed LI activations are not signaled via RADIUS to prevent that just the presence of CoA response NAK shows that LI request is not fake (action NOOP).

VSA 26-50058-140 - RtBrick-LI-Action (salt encrypted integer)

Value	Code	Description
NOOP	0	No action / Ignore LI request
ON	1	Start LI / Add LI request
OFF	2	Stop LI / Delete LI request

VSA 26-50058-141 - RtBrick-LI-Identifier (salt encrypted integer)

Device unique lawful interception identifier (LIID) within the range from 1 to 4194303.

VSA 26-50058-142 - RtBrick-LI-Direction (salt encrypted integer)

Value	Code	Description
INGRESS	1	Ingress mirroring only (from subscriber)
EGRESS	2	Egress mirroring only (to subscriber)
BOTH	3	Bidirectional mirroring (from and to subscriber)

VSA 26-50058-143 - RtBrick-LI-MED-Instance (salt encrypted string)

Routing instance through which the mediation device is reachable.

VSA 26-50058-144 - RtBrick-LI-MED-IP (salt encrypted IPv4 address)

IPv4 address of the mediation device.

VSA 26-50058-145 - RtBrick-LI-MED-Port (salt encrypted integer)

UDP port between 49152 and 65535 are set in the mirrored traffic.

RADIUS Framed Routes

The framed-route attributes are used by the RADIUS to install specific routes for a given subscriber. The corresponding next-hop information is automatically derived from the subscriber context. This functionality is typically used to route additional customer networks through the client device (CPE).

Those attributes are supported via RADIUS access-accept only. Each of those attributes may occur multiple times to install various routes. The actual amount of routes supported is limited by the maximum RADIUS packet size only.

The framed-route attributes are defined in RFC 2865 for IPv4 and RFC 3162 for IPv6 but the actual content of those attributes is implementation dependent. RBFS is expecting a string with a prefix followed by prefix length. The prefix can be followed by further routing options like costs, metric or even next-hop which are currently ignored.

Attribute 22 - Framed-Route

```
Framed-Route = "198.51.100.69/24"
```

Attribute 99 - Framed-IPv6-Route

```
Framed-IPv6-Route = "2001:db8:0:90::/32"
```

RADIUS Subscriber ACLs (Filters)

The subscriber ACL attributes are used by the RADIUS server to install a specific ACL, which has already been configured in the BNG, for a subscriber. For information about Subscriber ACL Configuration see 'Subscriber ACLs and HTTP Redirect Service' guide. After the Subscriber ACLs are configured with the name and the rules, use the following RADIUS attributes dynamically for adding or changing those ACLs from RADIUS. These attributes are supported in access-accept and CoA.

- ***VSA 26-50058-76 - RtBrick-IPv4-ACL-In***

  ```
  Attach IPv4 subscriber ingress ACL (ACL in upstream direction)
  ```

- ***VSA 26-50058-77 - RtBrick-IPv4-ACL-Out***

```
Attach IPv4 subscriber ingress ACL (ACL in downstream direction)
```

- ***VSA 26-50058-78 - RtBrick-IPv6-ACL-In***

  ```
  Attach IPv6 subscriber ingress ACL (ACL in upstream direction)
  ```

- ***VSA 26-50058-79 - RtBrick-IPv6-ACL-Out***

  ```
  Attach IPv6 subscriber egress ACL (ACL in downstream direction)
  ```

Example:
RtBrick-IPv4-ACL-In = “ipoe-sub1-http-acl“

TO detach any of the preceding ACLs and to delete the ACL associated with a subscriber, null string can be used with the corresponding attribute.

Example:
RtBrick-IPv4-ACL-In = ““

RADIUS HTTP Redirect URL

When HTTP Redirect is enabled for a subscriber either through Ascend-Data-Filter or through the subscriber ACLs, HTTP packets from the subscriber get redirected to the URL set by this attribute:

VSA 26-50058-75 - RtBrick-HTTP-Redirect-URL

This attribute is supported on RADIUS access-accept and CoA requests. This attribute allows the overwriting of the URL.

For example:

RtBrick-HTTP-Redirect-URL = “http://portal.rtbrick.com”

To reset the URL:

RtBrick-HTTP-Redirect-URL = “”

RADIUS Terminate Codes

VSA 26-50058-27 - RtBrick-Terminate-Code

The RtBrick-Terminate-Code is sent along with the standard Acct-Terminate-Cause (attribute 49). The list below shows the RtBrick termination codes with the mapping to the standard cause.

RtBrick-Terminate-Code	Acct-Terminate-Cause (RADIUS Attribute 49)	Description
0	10 (NAS Request)	NA
1	10 (NAS Request)	Subscriber Management Unknown Error
2	9 (NAS Error)	SubscriberD Internal Error
3	9 (NAS Error)	PPPoED Internal Error
4	9 (NAS Error)	PPPoED Object Deleted *This code is used if an PPPoE session object is deleted which is an indication for a crash or some other issues in PPPoE daemon.*
5	9 (NAS Error)	L2TPD Internal Error
6	9 (NAS Error)	L2TPD Object Deleted *This code is used if an L2TP session object is deleted which is an indication for a crash or some other issues in L2TP daemon.*
7	9 (NAS Error)	AAA Profile Not Found
8	9 (NAS Error)	RADIUS Profile Not Found
9	9 (NAS Error)	Authentication Type Missing
10	9 (NAS Error)	Authentication Order Missing or Invalid
11	10 (NAS Request)	Authentication Failure
12	10 (NAS Request)	Local Authentication Failed

RtBrick-Terminate-Code	Acct-Terminate-Cause (RADIUS Attribute 49)	Description
13	10 (NAS Request)	Accounting-Request-On Wait
14	9 (NAS Error)	No RADIUS Authentication Server Configured
15	10 (NAS Request)	RADIUS Authentication Failed
16	17 (User Error)	Authentication Rejected
17	17 (User Error)	Local Authentication Rejected
18	17 (User Error)	RADIUS Authentication Rejected
19	5 (Session Timeout)	Session Timeout
20	4 (Idle Timeout)	Idle Timeout
21	6 (Admin Reset)	Clear Session
22	6 (Admin Reset)	RADIUS CoA Disconnect
23	9 (NAS Error)	PPPoE Unknown Error
24	1 (User Request)	PPPoE PADT Received
25	9 (NAS Error)	PPPoE LCP Error
26	10 (NAS Request)	PPPoE LCP Generic Error
27	1 (User Request)	PPPoE LCP Terminate Request Received
28	10 (NAS Request)	PPPoE LCP Maximum Reject / NAK Exceeded
29	10 (NAS Request)	PPPoE LCP Negotiation Failed
30	10 (NAS Request)	PPPoE LCP Configuration-Request Exceeded
31	10 (NAS Request)	PPPoE LCP Echo-Request Timeout Exceeded
32	9 (NAS Error)	PPPoE PAP Error
33	9 (NAS Error)	PPPoE CHAP Error
34	9 (NAS Error)	PPPoE IPCP Error
35	9 (NAS Error)	PPPoE IP6CP Error

RtBrick-Terminate-Code	Acct-Terminate-Cause (RADIUS Attribute 49)	Description
36	9 (NAS Error)	PPPoE NCP Initialization Failed
37	10 (NAS Request)	PPPoE NCP Down
38	6 (Admin Reset)	PPPoE Clear Session
39	10 (NAS Request)	PPPoE Upper Layer Down
40	8 (Port Error)	PPPoE Interface Down
41	10 (NAS Request)	PPPoE Configuration Deleted
42	9 (NAS Error)	PPPoE Access Configuration Profile Not Found
43	9 (NAS Error)	L2TP Unknown Error
44	10 (NAS Request)	L2TP Tunnel Down
45	10 (NAS Request)	L2TP Tunnel Dead
46	10 (NAS Request)	L2TP Tunnel Deleted
47	10 (NAS Request)	L2TP No Tunnel Available
48	10 (NAS Request)	L2TP CDN Request
49	9 (NAS Error)	L2TP Profile Not Found
50	9 (NAS Error)	L2TP Access Configuration Profile Not Found
51	9 (NAS Error)	L2TP No Tunnel Pool Error
52	9 (NAS Error)	L2TP Local Tunnel Pool Error
53	9 (NAS Error)	L2TP RADIUS Tunnel Pool Error
54	6 (Admin Reset)	L2TP Clear Session
55	9 (NAS Error)	Access Configuration Profile Not Found
56	9 (NAS Error)	Local IPv4 Address Pool Not Found
57	10 (NAS Request)	Local IPv4 Address Pool Exhausted
58	9 (NAS Error)	Local IPv6 Prefix Pool Not Found

RtBrick-Terminate-Code	Acct-Terminate-Cause (RADIUS Attribute 49)	Description
59	10 (NAS Request)	Local IPv6 Prefix Pool Exhausted
60	9 (NAS Error)	Local IPv6 Delegated Prefix Pool Not Found
61	10 (NAS Request)	Local IPv6 Delegated Prefix Pool Exhausted
62	10 (NAS Request)	PPPoE CHAP Response Timeout
63	10 (NAS Request)	L2TP Session Deleted
64	10 (NAS Request)	Duplicate IPv4 address detected
65	10 (NAS Request)	Duplicate IPv6 prefix detected
66	10 (NAS Request)	Duplicate delegated IPv6 prefix detected
67	9 (NAS Error)	Routing instance not found
68	6 (Admin Reset)	Clear Session Force
69	6 (Admin Reset)	Test AAA Request Object Deleted
70	9 (NAS Error)	IPoED Object Deleted *This code is used if an IPoE subscriber object is deleted which is an indication for a crash or some other issues in IPoE daemon.*
71	9 (NAS Error)	IPoED Internal Error
72	9 (NAS Error)	IPoE Unknown Error
73	10 (NAS Request)	IPoE Upper Layer Down
74	10 (NAS Request)	IPoE Lower Layer Down
75	10 (NAS Request)	IPoE Interface Down
76	10 (NAS Request)	IPoE Configuration Deleted

RtBrick-Terminate-Code	Acct-Terminate-Cause (RADIUS Attribute 49)	Description
77	10 (NAS Request)	SubscriberD Request
78	10 (NAS Request)	L2BSA Service Deleted
79	10 (NAS Request)	L2BSA Service Added
80	6 (Admin Reset)	IPoE Clear Subscriber
81	6 (Admin Reset)	PPPoE Clear Session Force
82	9 (NAS Error)	IPoE Invalid Gateway IFL
83	9 (NAS Error)	FWD Invalid
84	9 (NAS Error)	FWD Failed
85	9 (NAS Error)	FWD Deleted
86	10 (NAS Request)	Instance Deleted
87	10 (NAS Request)	Redundancy Switchover
88	10 (NAS Request)	Redundancy Active Node Termination
89	10 (NAS Request)	Redundancy Active Node Down
90	9 (NAS Error)	Redundancy Invalid State
91	9 (NAS Error)	Redundancy Invalid State for Sync
92	9 (NAS Error)	Redundancy Validation Failed
93	10 (NAS Request)	Redundancy Config Deleted
94	9 (NAS Error)	Redundancy Stale Timer Expiry
95	9 (NAS Error)	IPv4 In ACL Validation Failed
96	9 (NAS Error)	IPv4 Out ACL Validation Failed
97	9 (NAS Error)	IPv6 In ACL Validation Failed
98	9 (NAS Error)	IPv6 Out ACL Validation Failed
99	9 (NAS Error)	IP Address Allocation Failed

RtBrick-Terminate-Code	Acct-Terminate-Cause (RADIUS Attribute 49)	Description
100	9 (NAS Error)	No IP Address *This code is used if a forwarding subscriber object is deleted which is an indication of a crash or some other issues in the forwarding infrastructure.*

It is recommended to raise operational alarms for every termination cause received with a value of 9 (NAS Error) for further investigations.

RADIUS CoA Error Handling

CoA requests should be retried a few times in the unlikely event of CoA NAK with error-cause (RFC 5176 attribute 101) other than session-context-not-found (503) or missing-attribute (402). There is no rollback of failed CoA requests to the former state but retrying the same request is supported because changes are implemented in a more declarative way.

4.2.4. RADIUS Accounting

RADIUS accounting servers handle accounting records for subscribers. The subscriber daemon transmits RADIUS Accounting-Start, Interim and Stop messages to these servers. Accounting is the process of tracking subscriber activity and network resource usage in a subscriber session. This includes the session time called time accounting and the number of packets and bytes transmitted during the session called volume accounting.

A RADIUS Acct-Status-Type attribute is used by the RADIUS client (subscriber daemon) to mark the start of accounting (for example, upon booting) by specifying Accounting-On and to mark the end of accounting (for example, just before a scheduled reboot) by specifying Accounting-Off. This message is often used by RADIUS servers to automatically close/terminate all open accounting records/sessions for the corresponding client and therefore must not be sent to servers belonging to a profile which was already used/started for accounting.

Per default, the assumption is that all servers referenced by a RADIUS profile share the same states and therefore accounting-on must be only sent to one of those before the first accounting-start is sent.

RADIUS Accounting-On/Off messages are disabled by default and can be optionally enabled in the RADIUS profile configuration. There is also an additional configuration option to optionally wait for Accounting-On response which prevents any new session until accounting has started meaning that Accounting-On response is received.

ℹ Accounting-Off is currently not implemented!

RADIUS accounting requests are often used for billing and therefore should be able to store and retry over a longer period (common up to 24 hours or more) which can be optionally enabled in the RADIUS profile configuration using the accounting backup configuration option.

Time Accounting

All accounting relevant timestamps are retrieved using the UNIX system call clock_gettime internally stored in the datastore as 64 bit microseconds timestamp with the lower order 32 bit representing the seconds since January 1, 1970 00:00 UTC which is also known as epoch or unix timestamp and the higher order 32 bit representing the microseconds which are per definition less than 1.000.000 (one second). The seconds part is counted in full seconds which is similar to always rounded down.

The RADIUS attribute event-timestamp (type 55) defined in RFC2869 is included in each RADIUS Accounting-Request packet to record the time that this event occurred in seconds since January 1, 1970 00:00 UTC. A max allowed deviation of 500 milliseconds of a timestamp in seconds requires mathematical rounding of the internal 64 bit microseconds timestamps to the 32 bit RADIUS event-timestamp in seconds. This means to round down if the microseconds part of the timestamp is less than 0.5 seconds and rounded up if equal or greater than 0.5 seconds.

Start Timestamps

In RBFS the timestamp indicating the PPPoE session start will be generated after successful negotiation of at least one of the PPP network control protocols (IPCP for IPv4 or IP6CP for IPv6). In case of L2TP tunneled sessions (LAC), the successful L2TP session setup after sending ICCN is used as a start timestamp. In both cases this is also the trigger for the RADIUS accounting-request start where this timestamp is used as RADIUS event-timestamp.

Stop Timestamps

The timestamp indicating the session stop will be generated more or less immediately after termination request (e.g. timeout, user request, nas request, …).

The RADIUS attribute Acct-Session-Time (type 46) defined in RFC2866 indicates how many seconds the user has received service and can only be present in Accounting-Request records where the Acct-Status-Type is set to Interim or Stop. This time will be calculated based on the internal session start and stop time in microseconds and mathematical rounded to seconds. For RADIUS accounting interim requests the current event time is used instead of stop time.

> ℹ RFC and draft compliance are partial except as specified.

Volume Accounting

Based on §45g of the German Telecommunications Act (TKG), the maximum deviation for accounting relevant counters must be less than 1% per billing period which is typically one month.

The whole RBFS architecture is based on an event-based publish and subscribe model using the BDS infrastructure. Each process publishes all information and states to BDS and interested processes subscribe to that information. The whole inter process communications (IPC) is replaced by BDS table operations which allows to build a hyper scaled distributed software system. This means related to accounting that each counter is requested from data plane and updated in BDS unsolicited based on configurable intervals typically set between 5 to 30 seconds. This applies to any type of counters like interface counters or QoS statistics. This means that CLI commands or API requests will return the last updated counter and not the counter of the time of request. Therefore each BDS object contains a timestamp indicating the time of the last counter update which allows to use these counters with the required accuracy.

Subscriber volume accounting is based on multiple sources like logical interface (LIF), class/queue-, policer- and control-traffic statistics.

All those counters are layer 2 (L2) counters and some of them may reset through configuration changes like QoS counters after applying a new QoS profile. Therefore the subscriber management application daemon is doing some post processing of all counters to ensure that no accounting information is lost and to calculate the final accounting values. This includes also optional configurable counter adjustments.

There is a BDS database object in the table **local.access.subscriber.accounting** created for each subscriber which stores all counters and attributes required to calculate the actual volume counters. This object is created together with the subscriber and automatically deleted if the actual subscriber object is removed.

The volume accounting counters must not reset if the actual hardware counter is deleted or has changed/reset to zero. Therefore instead of using the source counter directly, the delta since the last interval is calculated and added to the final counter.

The usage of callback functions to counter delete and change events ensures that no accounting relevant information will be lost through reconfiguration of subscriber sessions.

The function which removes the subscriber counters from the data plane will receive the final counters before removal and update them into BDS counter objects before this is deleted.

Interims Volume Counters

RADIUS accounting interim requests will also use the last updated counters with current time as event-timestamp.

Stop Volume Counters

The session termination might be triggered by REST API, CLI (clear subscriber …) RADIUS CoA disconnect request, session timeout, idle timeout or client request (PADT). The corresponding RADIUS accounting stop requests will be delayed to wait for the final counter update but uses the timestamp of the terminate request.

Interims Accounting

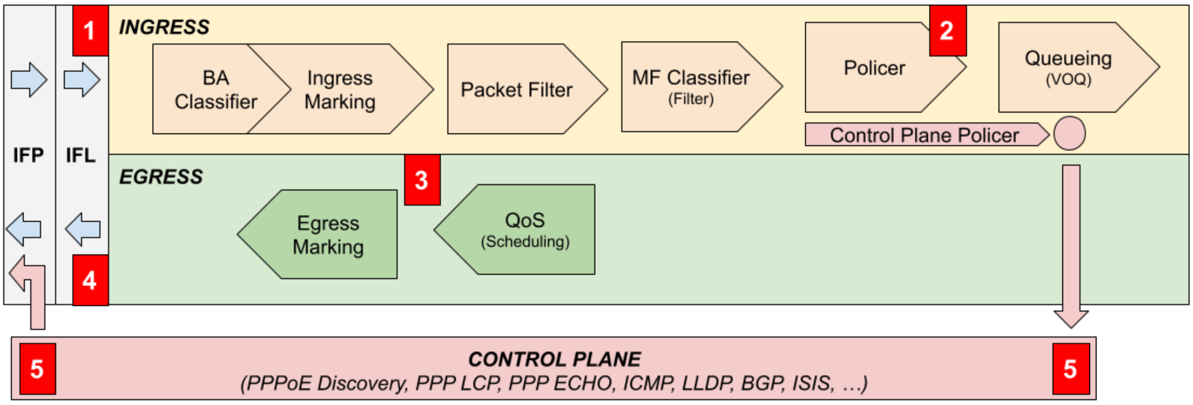
To receive RADIUS counters in fast intervals the corresponding session interim accounting interval can be set depending on the actual needs to any value between one second and multiple hours or days (recommended is at least 30 seconds). This interval can be also updated via CoA request using the Acct-Interim-Interval attribute which triggers an interim accounting immediately and uses the new interval from now onwards. This can be also used to request interim accounting requests on demand. Sending a CoA request with the applied interval triggers an accounting request but the original interval is not changed.

4.2.5. RADIUS Counters

The packets and byte counters of each session, traffic class (class 0 - 7) as well as policer counters (level 1 to 4) are supported via RADIUS accounting interims and stop messages. Those counters are layer two counters per default which can be adjusted using correction values and factors from the AAA configuration profile.

Subscriber accounting is based on multiple sources like LIF-, class/queue-, policer- and control-traffic statistics which are described below.

- The InLIF (Incoming Logical Interface) stats count all traffic received on the logical interface including control traffic are traffic dropped by ingress policer.
- The policer stats count all traffic accepted by policer (color green) per level (1-4). Ingress control traffic will be hit by a separate control plane policer and therefore not counted in the session policer stats.
- The class/queue stats count all egress traffic except control traffic which is directly sent to the IFP.
- The OutLIF (Outgoing Logical Interface) stats count all traffic sent on the logical interface except control traffic which is directly sent to the IFP.
- The control stats count all traffic received and sent from or to the control plane for a given subscriber (counted in PPPoE daemon for PPPoE sessions).



Because counter updates are not atomic operations, the sum of all class counters might be different from the session counters.

The counted bytes per packet can be adjusted as described in chapter Configuring Accounting Adjustments of the Subscriber Management Configuration Guide.

All subscriber accounting attributes and counters are stored without adjustment (L2) in the table **local.access.subscriber.accounting** and remain until the subscriber session is finally removed after response to RADIUS accounting stop.

The command show subscriber <id> accounting displays the adjusted subscriber accounting information. It is also possible to display the values before adjustment using show subscriber <id> accounting origin.

```
supervisor@rtbrick: op> show subscriber 72339069014638637 accounting
Subscriber-Id: 72339069014638637
    IFL: ppp-0/0/1/72339069014638637
    Start Timestamp: Wed Feb 24 09:06:47 GMT +0000 2021
    Idle Timestamp: Wed Feb 24 09:06:47 GMT +0000 2021
    Session-Timeout: 86400 seconds
    Idle-Timeout: 7200 seconds
    Session Statistics:
        Ingress: 0 packets 0 bytes
        Egress: 0 packets 0 bytes
    LIF Statistics:
        Ingress: 14 packets 896 bytes
        Egress: 0 packets 0 bytes
    Egress Class (Queue) Statistics:
        class-0: 0 packets 0 bytes
        class-1: 0 packets 0 bytes
        class-2: 0 packets 0 bytes
        class-3: 0 packets 0 bytes
        class-4: 0 packets 0 bytes
        class-5: 0 packets 0 bytes
        class-6: 0 packets 0 bytes
        class-7: 0 packets 0 bytes
    Ingress Policer Statistics:
        Level 1: 0 packets 0 bytes
        Level 2: 0 packets 0 bytes
        Level 3: 0 packets 0 bytes
        Level 4: 0 packets 0 bytes
```

Session Counters

Per default, the session counters are calculated using the LIF statistics which include all traffic received on the logical interface (InLIF) and all traffic sent on the logical interface (OutLIF) except control traffic which is directly sent to the IFP. Ingress traffic sent to CPU or transit traffic dropped by policer is still counted whereas egress traffic dropped by QoS is not counted.

Alternatively, it is possible to use the sum of all policer counters for ingress session counters which can be changed in the AAA configuration profile by setting the accounting ingress source to POLICER (default is LIF) to include only accepted transit traffic.

Using LIF in egress and POLICER in ingress results in a symmetric behavior where only accepted transit traffic is counted.

Following the list of the RtBrick class counter attributes.

Attribute	Name	Description
42	Acct-Input-Octets	Incoming session bytes (uint32)
43	Acct-Output-Octets	Outgoing session bytes (uint32)
47	Acct-Input-Packets	Incoming session packets (uint32)
48	Acct-Output-Packets	Outgoing session packets (uint32)
52	Acct-Input-Gigawords	Acct-Input-Octets overflow counter (uint32)
53	Acct-Output-Gigawords	Acct-Output-Octets overflow counter (uint32)

The internal 64 bit counters are split over the standard 32 bit octet and gigaword counters by using the most significant 32 bit as gigawords and the least significant 32 bits as octets.

32 Bit Acct-Input/Output-Gigawords	32 Bit Acct-Input/Output-Octets
Internal 64 Bit Counter	

Class Counters

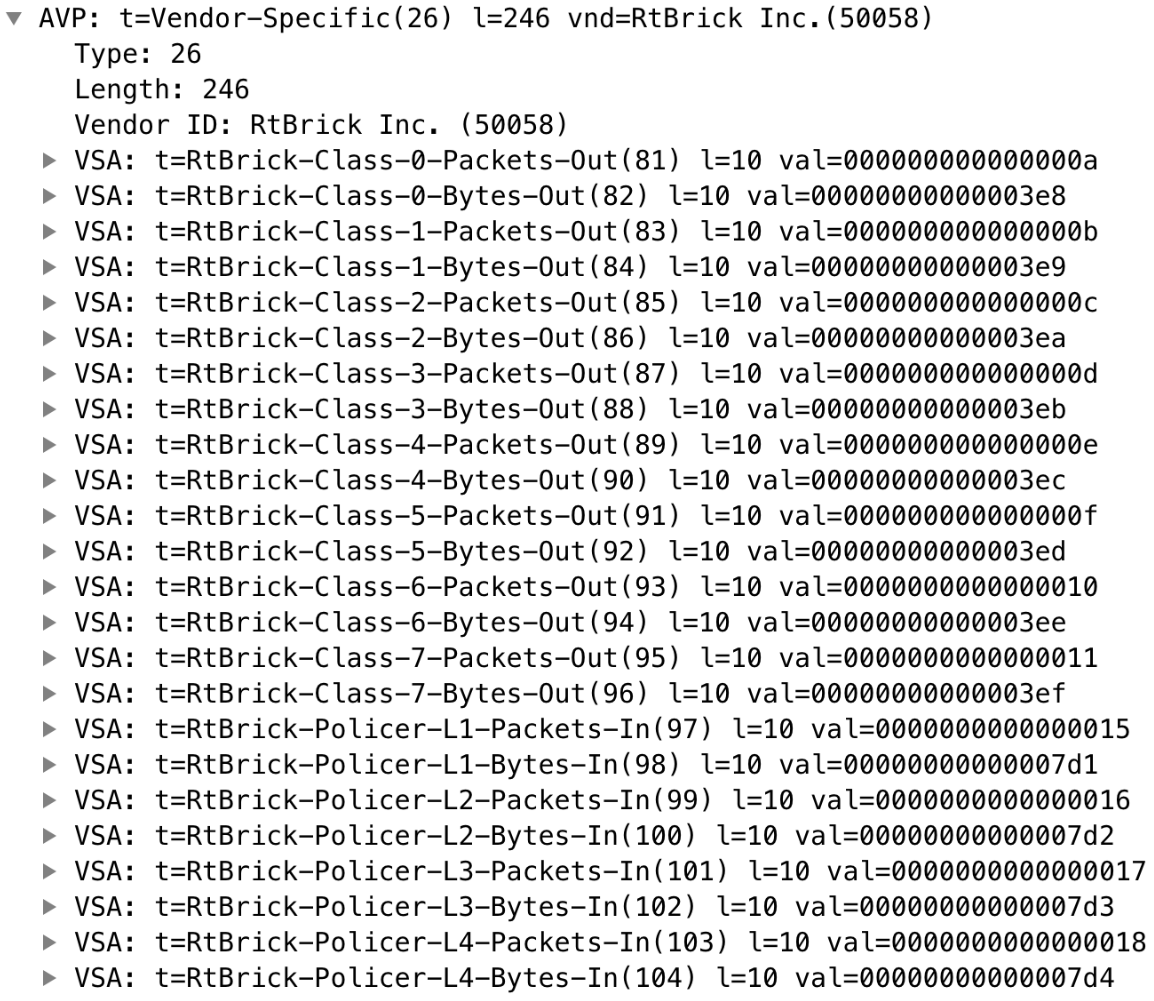
The outgoing class counters are filled by queue counters which ensures that only traffic leaving the device is counted here.

Following the list of the RtBrick class counter attributes.

Attribute	Name	Description
26-50058-81	RtBrick-Class-0-Packets-Out	Outgoing Class-0 packets (uint64)
26-50058-82	RtBrick-Class-0-Bytes-Out	Outgoing Class-0 bytes (uint64)

Attribute	Name	Description
26-50058-83	RtBrick-Class-1-Packets-Out	Outgoing Class-1 packets (uint64)
26-50058-84	RtBrick-Class-1-Bytes-Out	Outgoing Class-1 bytes (uint64)
26-50058-85	RtBrick-Class-2-Packets-Out	Outgoing Class-2 packets (uint64)
26-50058-86	RtBrick-Class-2-Bytes-Out	Outgoing Class-2 bytes (uint64)
26-50058-87	RtBrick-Class-3-Packets-Out	Outgoing Class-3 packets (uint64)
26-50058-88	RtBrick-Class-3-Bytes-Out	Outgoing Class-3 bytes (uint64)
26-50058-98	RtBrick-Class-4-Packets-Out	Outgoing Class-4 packets (uint64)
26-50058-90	RtBrick-Class-4-Bytes-Out	Outgoing Class-4 bytes (uint64)
26-50058-91	RtBrick-Class-5-Packets-Out	Outgoing Class-5 packets (uint64)
26-50058-92	RtBrick-Class-5-Bytes-Out	Outgoing Class-5 bytes (uint64)
26-50058-93	RtBrick-Class-6-Packets-Out	Outgoing Class-6 packets (uint64)
26-50058-94	RtBrick-Class-6-Bytes-Out	Outgoing Class-6 bytes (uint64)
26-50058-95	RtBrick-Class-7-Packets-Out	Outgoing Class-7 packets (uint64)
26-50058-96	RtBrick-Class-7-Bytes-Out	Outgoing Class-7 bytes (uint64)

Those attributes will be encoded as subattributes (RFC2865) similar to broadband forum (BBF) line attributes to reduce the size of the RADIUS message. Counters with a zero value will be also excluded from the packet.



```
▼ AVP: t=Vendor-Specific(26) l=246 vnd=RtBrick Inc.(50058)
    Type: 26
    Length: 246
    Vendor ID: RtBrick Inc. (50058)
  ▶ VSA: t=RtBrick-Class-0-Packets-Out(81) l=10 val=000000000000000a
  ▶ VSA: t=RtBrick-Class-0-Bytes-Out(82) l=10 val=00000000000003e8
  ▶ VSA: t=RtBrick-Class-1-Packets-Out(83) l=10 val=000000000000000b
  ▶ VSA: t=RtBrick-Class-1-Bytes-Out(84) l=10 val=00000000000003e9
  ▶ VSA: t=RtBrick-Class-2-Packets-Out(85) l=10 val=000000000000000c
  ▶ VSA: t=RtBrick-Class-2-Bytes-Out(86) l=10 val=00000000000003ea
  ▶ VSA: t=RtBrick-Class-3-Packets-Out(87) l=10 val=000000000000000d
  ▶ VSA: t=RtBrick-Class-3-Bytes-Out(88) l=10 val=00000000000003eb
  ▶ VSA: t=RtBrick-Class-4-Packets-Out(89) l=10 val=000000000000000e
  ▶ VSA: t=RtBrick-Class-4-Bytes-Out(90) l=10 val=00000000000003ec
  ▶ VSA: t=RtBrick-Class-5-Packets-Out(91) l=10 val=000000000000000f
  ▶ VSA: t=RtBrick-Class-5-Bytes-Out(92) l=10 val=00000000000003ed
  ▶ VSA: t=RtBrick-Class-6-Packets-Out(93) l=10 val=0000000000000010
  ▶ VSA: t=RtBrick-Class-6-Bytes-Out(94) l=10 val=00000000000003ee
  ▶ VSA: t=RtBrick-Class-7-Packets-Out(95) l=10 val=0000000000000011
  ▶ VSA: t=RtBrick-Class-7-Bytes-Out(96) l=10 val=00000000000003ef
  ▶ VSA: t=RtBrick-Policer-L1-Packets-In(97) l=10 val=0000000000000015
  ▶ VSA: t=RtBrick-Policer-L1-Bytes-In(98) l=10 val=00000000000007d1
  ▶ VSA: t=RtBrick-Policer-L2-Packets-In(99) l=10 val=0000000000000016
  ▶ VSA: t=RtBrick-Policer-L2-Bytes-In(100) l=10 val=00000000000007d2
  ▶ VSA: t=RtBrick-Policer-L3-Packets-In(101) l=10 val=0000000000000017
  ▶ VSA: t=RtBrick-Policer-L3-Bytes-In(102) l=10 val=00000000000007d3
  ▶ VSA: t=RtBrick-Policer-L4-Packets-In(103) l=10 val=0000000000000018
  ▶ VSA: t=RtBrick-Policer-L4-Bytes-In(104) l=10 val=00000000000007d4
```

Policer Counters

The incoming policer attributes count all traffic accepted (colored green) by ingress policers with dedicated packet and byte values per level.

Following the list of the RtBrick policer counter attributes.

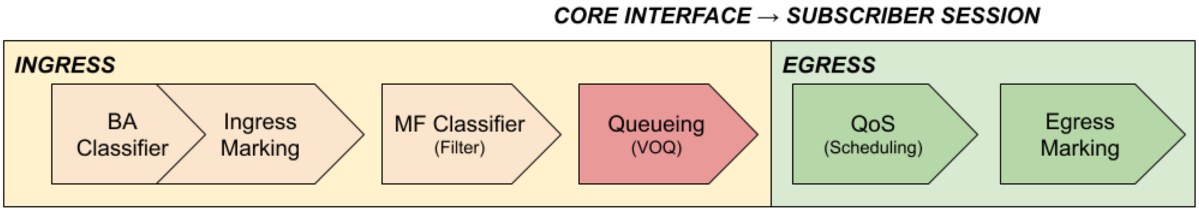
Attribute	Name	Description
26-50058-97	RtBrick-Policer-L1-Packets-In	Incoming Policer L1 accepted packets (uint64)
26-50058-98	RtBrick-Policer-L1-Bytes-In	Incoming Policer L1 accepted bytes (uint64)
26-50058-99	RtBrick-Policer-L2-Packets-In	Incoming Policer L2 accepted packets (uint64)
26-50058-100	RtBrick-Policer-L2-Bytes-In	Incoming Policer L2 accepted bytes (uint64)

Attribute	Name	Description
26-50058-101	RtBrick-Policer-L3-Packets-In	Incoming Policer L3 accepted packets (uint64)
26-50058-102	RtBrick-Policer-L3-Bytes-In	Incoming Policer L3 accepted bytes (uint64)
26-50058-103	RtBrick-Policer-L4-Packets-In	Incoming Policer L4 accepted packets (uint64)
26-50058-104	RtBrick-Policer-L4-Bytes-In	Incoming Policer L4 accepted bytes (uint64)

Those attributes will be encoded as subattributes (RFC2865) similar to broadband forum (BBF) line attributes to reduce the size of the RADIUS message. Counters with a zero value will be also excluded from the packet.

Service Classification and Accounting

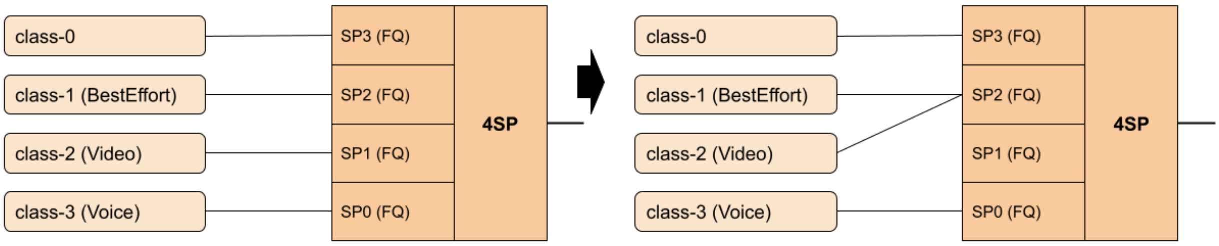
The underlying virtual output queueing (VOQ) is a common and efficient architecture for high performance switches but brings some major limitations. One obvious limitation is that queues must be allocated in ingress and canΓÇÖt be changed in egress as shown in the picture below.



Service classification and accounting is based on standard behavior aggregate (BA) and multifield classifier (MF) bound global or to ingress interfaces matching on some protocol fields assigning a traffic class. Downstream traffic (to subscriber) is counted per subscriber and class using queue counters to count only traffic not dropped by QoS. Upstream traffic (from subscribers) is countered per subscriber and policer level.

Some service providers would like to control if premium traffic is handled differently on a per subscriber basis which can be also achieved with VOQ as explained below.

Let us assume a product which handles video streaming services with higher priority or excludes those traffic from actual data volume. With VOQ the classification as Video is done on the ingress core interface equally for all subscribers regardless of the booked product. Now instead of changing the queue per subscriber in egress, the behavior of the queue can be changed to behave equally to BestEffort which is supported in egress per subscriber.



The corresponding Video traffic is now threaded equally to BestEffort but still counted separately. For accounting purposes the traffic of BestEffort and Video can be simply added in the billing infrastructure to get the actual amount of BestEffort traffic.

In upstream, the expected class and policer level can be assigned per subscriber because BA and MF classifiers are bound to the PPPoE session in ingress.

4.2.6. RADIUS Accounting Adjustment Configuration

The accounting adjustment allows to do some basic counter adjustment for RADIUS interims and stop accounting request messages using the following parameters from the AAA configuration profile (**global.access.aaa.profile.config**).

This counter adjustment allows normalizing counters with different encapsulations (double tagged, untagged, …) to L3 counters for example.

Parameter	Type	Description
accounting_ingress_source	uint8	Source of session ingress counter (1: LIF or 3: POLICER) *Default: LIF*
accounting_egress_source	uint8	Source of session egress counter (1: LIF or 2: CLASS) *Default: LIF*

Parameter	Type	Description
ingress_byte_adjustment_value	float	Adjust ingress LIF counters by +/-N bytes per packet *Default: 0.00*
ingress_byte_adjustment_factor	float	Adjust ingress LIF counters by factor (executed after adjustment value) *Default: 1.00*
egress_byte_adjustment_value	float	Adjust egress LIF counters by +/-N bytes per packet *Default: 0.00*
egress_byte_adjustment_factor	float	Adjust egress LIF counters by factor (executed after adjustment value) *Default: 1.00*
accounting_egress_class_byte_adjustment_value	float	Adjust egress CLASS counters by +/-N bytes per packet *Default: 0.00*
accounting_egress_class_byte_adjustment_factor	float	Adjust egress POLICER counters by factor (executed after adjustment value) *Default: 1.00*
accounting_ingress_policer_byte_adjustment_value	float	Adjust ingress POLICER counters by +/-N bytes per packet *Default: 0.00*
accounting_ingress_policer_byte_adjustment_factor	float	Adjust ingress LIF counters by factor (executed after adjustment value) *Default: 1.00*

The byte adjustment value supports positive and negative values like -20.0 or 20.0. Provided decimal digits in the adjustment values are ignored.

The byte adjustment factors support positive values and only the first two decimal digits are used like 0.98 (-2%) or 1.02 (+2%).

4.2.7. RADIUS Redundancy

It is possible to configure multiple RADIUS authentication and accounting servers for redundancy and or load-balancing.

The following two algorithms are supported:

- **DIRECT (default):** Requests are sent to the server following the one where the last request was sent. If the subscriber daemon receives no response from the server, requests are sent to the next server and so on.
- **ROUND-ROBIN:** Requests are sent to the server following the one where the last request was sent. If the subscriber daemon router receives no response from the server, requests are sent to the next server and so on.

4.2.8. Test AAA

The test AAA subscriber feature allows operators to verify and test authentication and accounting (e.g. local or RADIUS) by emulating a subscriber without the need for external clients to be connected. This is a commonly used feature used during troubleshooting or to validate the RADIUS configuration.

Test subscribers will be created by adding a request object into the test aaa request table (**global.subscriber.1.test.aaa.request**) via API or CLI. This request object includes beside username and password also a spoofed interface, outer-vlan, inner-vlan and MAC address required to build corresponding attributes like NAS port identifiers or the vendor specific RtBrick-Access-Stack. It is also possible to add an Agent-Circuit-Id or Agent-Remote-Id to test line based authentication.

This new object will first trigger a validation plugin to reject invalid requests (e.g. subscriber-id out of range, missing mandatory attributes, …). The add callback for this object will trigger the creation of the actual subscriber. Deleting the request object will trigger the termination of the subscriber. If this subscriber terminates for other reasons like CoA or CLI, the subscriber remains in a terminated state until the request object is deleted.

Compared to real subscribers, test subscribers will not trigger any forwarding state. Therefore no object will be created in the following tables for those subscribers:

- global.access.1.subscriber.ifl

But the following tables will be populated for operators to check dynamic QoS, filters or IGMP:

- global.access.1.subscriber.qos
- global.access.1.subscriber.qos.shaper
- global.access.1.subscriber.filter
- global.access.1.subscriber.igmp.service

Those entries are ignored internally as there is no matching subscriber interface (IFL) (global.access.1.subscriber.ifl). The only exception here is the filter table meaning that received filters (ADF) learned from RADIUS are installed in the ACL table and applied with the IP addresses of the test subscribers.

Test AAA Subscriber-Id:

In RBFS each subscriber is uniquely identified by subscriber-id which is a 64 bit number allocated in a distributed fashion by the application creating the subscriber (for example, pppoed for pppoe sessions). This uniqueness is achieved with the following format.

The first 8 bits are used to identify the application:

- 0x00 subscriberd
- 0x01 pppoed
- 0x02 l2tpd

The next 8 bits are used to identify the application sharding instance allocated (for example, pppoed.1, pppoed.2, and so on).

The remaining 48 bits are used to uniquely identify the subscriber, which theoretically allows up to 281474976710656 subscribers per application instance.

An emulated test subscriber needs also a subscriber-id.

```
-----
* 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

* +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

* | app-id (0) | app-instance | reserved |

* +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

* | subscriber

* +-+-+-+-+

-----
```

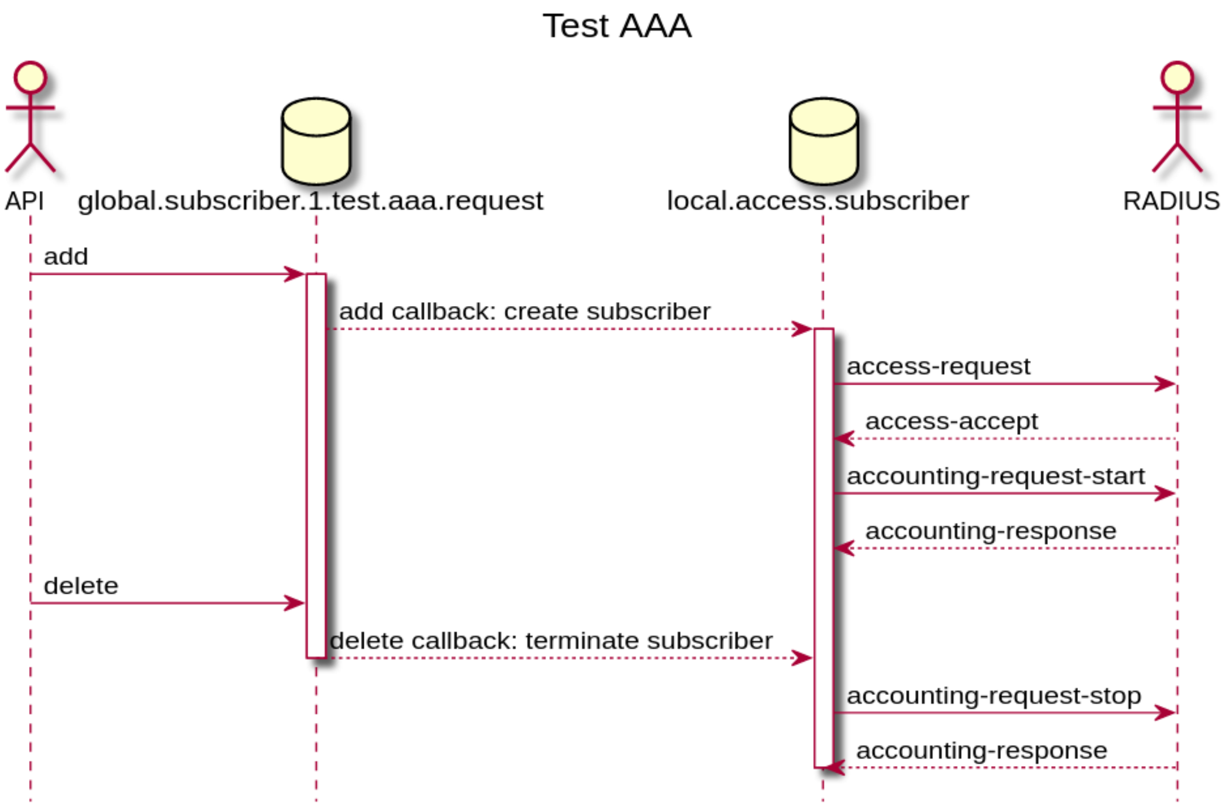
Assuming subscriber daemon instance 1 (subscriberd.1), the valid range for test subscriber identifiers is between 281474976710656 and 281479271677951.

```
Min Subscriber-Id: 281474976710656
0000 0000 0000 0001 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

Max Subscriber-Id: 281479271677951
0000 0000 0000 0001 0000 0000 0000 0000 1111 1111 1111 1111 1111 1111 1111 1111
```

The subscriber identifier will be allocated manually by operators from this range which has the advantage to easier identify the subscriber created by request.

Test AAA RADIUS Flow:



Test AAA Attributes

Attribute	Mandatory	API	CLI	Default
Subscriber-Id	Yes	subscriber_id	subscriber-id	
Access Profile	Yes	access_profile_name	access-profile	
AAA Profile	Yes	aaa_profile_name	aaa-profile	
Username	No	user_name	username	test
Password	No	password	password	test
Agent-Circuit-id	No	aci	aci	
Agent-Remote-id	No	ari	ari	
Interface (IFP)	*	interface_name	interface	ifp-0/0/0
Outer VLAN	No	outer_vlan	outer-vlan	0
Inner VLAN	No	inner_vlan	inner-vlan	0

Attribute	Mandatory	API	CLI	Default
MAC Address	No	client_mac	**	00:00:00:00:00:00

** The interface is mandatory via CLI but optional via API.*

*** The client MAC address is currently supported via API only.*

Test AAA via CLI

Note: In CLI strings must not contain the ΓÇ£#ΓÇ¥ and therefore this is not permitted for username, aci or ari set via CLI. This limitation is valid for CLI only as subscribers created via API are allowed to use this the ΓÇ£#ΓÇ¥ in the mentioned strings.

Add Test Subscriber:

```
supervisor@leaf1: op> test subscriber aaa request 281474976710656 pppoe-dual aaa-
default ifl-0/0/1 username user1@rtbrick.com aci "0.0.0.0/0.0.0.0 eth 0/0" ari
DEU.RTBRICK.01
```

List All Test Subscribers:

```
supervisor@leaf1: op> show subscriber test
Subscriber-Id          Interface        VLAN      State            Username
281474976710656        ifp-0/1/23     0:0       ESTABLISHED
user1@rtbrick.com
```

Delete Test Subscriber:

```
supervisor@leaf1: op> test subscriber aaa delete 281474976710656
```

Test AAA via API

Add Test Subscriber:

```
{{rbfs_url}}/api/v1/rbfs/elements/{\{element}}/services/subscriberd.1/proxy/bds/ob
ject/add
```

```
{
   "table": {
       "table_name": "global.access.1.test.aaa.request"
   },
   "objects": [
       {
           "attribute": {
               "subscriber_id": 281474976710658,
               "aaa_profile_name": "aaa-default",
               "access_profile_name": "pppoe-dual",
               "user_name": "user@rtbrick.com",
               "ari": "DEU.RTBRICK.01",
               "aci": "0.0.0.0/0.0.0.0 eth 0/0"
           }
       }
   ]
}
```

Delete Test Subscriber:

```
{{rbfs_url}}/api/v1/rbfs/elements/{\{element}}/services/subscriberd.1/proxy/bds/object/delete
```

```
{
   "table": {
       "table_name": "global.access.1.test.aaa.request"
   },
   "objects": [
       {
           "attribute": {
               "subscriber_id": 281474976710658
           }
       }
   ]
}
```

List All Test Subscribers:

```
{{rbfs_url}}/api/v1/rbfs/elements/{\{element}}/services/subscriberd.1/proxy/bds/table/walk
```

```
{
   "table": {
       "table_name": "global.access.1.test.aaa.request"
   }
}
```

4.2.9. Appendix A - RADIUS Dictionary

Following the RtBrick RADIUS dictionary in the well known FreeRADIUS format.

```
# This dictionary applies to RtBrick Full Stack (RBFS)
# https://www.rtbrick.com/
#
VENDOR          RtBrick 50058

BEGIN-VENDOR    RtBrick

ATTRIBUTE       RtBrick-Access-Hostname              21  string
ATTRIBUTE       RtBrick-Access-Port                  22  string
ATTRIBUTE       RtBrick-Access-Stack                 23  string
ATTRIBUTE       RtBrick-Access-MAC-Address           24  string

ATTRIBUTE       RtBrick-Subscriber-Id                25  integer64
ATTRIBUTE       RtBrick-Subscriber-Ifl               26  string

ATTRIBUTE       RtBrick-Terminate-Code               27  integer

ATTRIBUTE       RtBrick-L2TP-Pool                    40  string
ATTRIBUTE       RtBrick-L2TP-Profile                 41  string
ATTRIBUTE       RtBrick-L2TP-Tx-Connect-Speed        42  integer
ATTRIBUTE       RtBrick-L2TP-Rx-Connect-Speed        43  integer

ATTRIBUTE       RtBrick-QoS-Profile                  62  string
ATTRIBUTE       RtBrick-QoS-Shaper                   63  string
ATTRIBUTE       RtBrick-QoS-Parent-Scheduler         64  string
ATTRIBUTE       RtBrick-QoS-Policer                  65  string
ATTRIBUTE       RtBrick-QoS-MFC                      66  string
ATTRIBUTE       RtBrick-QoS-Queues                   67  string

ATTRIBUTE       RtBrick-AAA-Profile                  69  string
ATTRIBUTE       RtBrick-Service-Profile              70  string

ATTRIBUTE       RtBrick-IGMP-Status                  71  integer
ATTRIBUTE       RtBrick-IGMP-Profile                 72  string
ATTRIBUTE       RtBrick-IGMP-Version                 73  integer
ATTRIBUTE       RtBrick-IGMP-Max-Members             74  integer

ATTRIBUTE       RtBrick-IPv4-ACL-In                  76  string
ATTRIBUTE       RtBrick-IPv4-ACL-OUT                 77  string
ATTRIBUTE       RtBrick-IPv6-ACL-In                  78  string
ATTRIBUTE       RtBrick-IPv6-ACL-Out                 79  string

ATTRIBUTE       RtBrick-Class-0-Packets-Out          81  integer64
ATTRIBUTE       RtBrick-Class-0-Bytes-Out            82  integer64
ATTRIBUTE       RtBrick-Class-1-Packets-Out          83  integer64
ATTRIBUTE       RtBrick-Class-1-Bytes-Out            84  integer64
ATTRIBUTE       RtBrick-Class-2-Packets-Out          85  integer64
ATTRIBUTE       RtBrick-Class-2-Bytes-Out            86  integer64
ATTRIBUTE       RtBrick-Class-3-Packets-Out          87  integer64
ATTRIBUTE       RtBrick-Class-3-Bytes-Out            88  integer64
ATTRIBUTE       RtBrick-Class-4-Packets-Out          89  integer64
ATTRIBUTE       RtBrick-Class-4-Bytes-Out            90  integer64
ATTRIBUTE       RtBrick-Class-5-Packets-Out          91  integer64
ATTRIBUTE       RtBrick-Class-5-Bytes-Out            92  integer64
ATTRIBUTE       RtBrick-Class-6-Packets-Out          93  integer64
ATTRIBUTE       RtBrick-Class-6-Bytes-Out            94  integer64
ATTRIBUTE       RtBrick-Class-7-Packets-Out          95  integer64
ATTRIBUTE       RtBrick-Class-7-Bytes-Out            96  integer64

ATTRIBUTE       RtBrick-Policer-L1-Packets-In        97  integer64
ATTRIBUTE       RtBrick-Policer-L1-Bytes-In          98  integer64
ATTRIBUTE       RtBrick-Policer-L2-Packets-In        99  integer64
ATTRIBUTE       RtBrick-Policer-L2-Bytes-In         100  integer64
ATTRIBUTE       RtBrick-Policer-L3-Packets-In       101  integer64
ATTRIBUTE       RtBrick-Policer-L3-Bytes-In         102  integer64
ATTRIBUTE       RtBrick-Policer-L4-Packets-In       103  integer64
ATTRIBUTE       RtBrick-Policer-L4-Bytes-In         104  integer64

ATTRIBUTE       RtBrick-DNS-Primary                 131  ipaddr
ATTRIBUTE       RtBrick-DNS-Secondary               132  ipaddr
ATTRIBUTE       RtBrick-DNS-Primary-IPv6            134  ipv6addr
ATTRIBUTE       RtBrick-DNS-Secondary-IPv6          135  ipv6addr

ATTRIBUTE       RtBrick-Instance                    137  string

ATTRIBUTE       RtBrick-Connection-Status-Message   139  string

ATTRIBUTE       RtBrick-LI-Action                   140  integer encrypt=2
ATTRIBUTE       RtBrick-LI-Identifier               141  integer encrypt=2
ATTRIBUTE       RtBrick-LI-Direction                142  integer encrypt=2
ATTRIBUTE       RtBrick-LI-MED-Instance             143  string  encrypt=2
ATTRIBUTE       RtBrick-LI-MED-IP                   144  ipaddr  encrypt=2
ATTRIBUTE       RtBrick-LI-MED-Port                 145  integer encrypt=2

#         VALUE MAPS
#
#       Attribute               Name        Number
VALUE   RtBrick-LI-Action       NOOP        0
VALUE   RtBrick-LI-Action       ON          1
VALUE   RtBrick-LI-Action       OFF         2

VALUE   RtBrick-LI-Direction    INGRESS     1
VALUE   RtBrick-LI-Direction    EGRESS      2
VALUE   RtBrick-LI-Direction    BOTH        3

VALUE   RtBrick-IGMP-Status     DISABLED    0
VALUE   RtBrick-IGMP-Status     ENABLED     1

VALUE   RtBrick-IGMP-Version    V1          1
VALUE   RtBrick-IGMP-Version    V2          2
VALUE   RtBrick-IGMP-Version    V3          3

END-VENDOR RtBrick
```

4.2.10. Appendix B - RADIUS Standard Attributes Supported

Disclaimer Note: The following table presents the current development state of the RADIUS attributes supported in RBFS. Some of these may be incomplete and are subjected to change at any time without any notice.

RFC	Attribute		Access		Accounting				CoA	
	#	Name	Request	Response	Acct-Start	Acct-Interim	Acct-Stop	Acct-On	Request	Response

2865	

	Message								
25	Class		X	X	X	X			
32	NAS-Identifier	X		X	X	X	X		
61	NAS-Port-Type	X		X	X	X			

2688	40	Acct-Status-Type			X	X	X	X		
	41	Acct-Delay-Time								
	42	Acct-Input-Octets				X	X			
	43	Acct-Output-Octets				X	X			
	44	Acct-Session-ID	X		X	X	X			
	45	Acct-Authentic								
	46	Acct-Session-Time					X			
	47	Acct-Input-Packets				X	X			
	48	Acct-Output-Packets				X	X			
	49	Acct-Terminate-Cause					X			

2869	52	Acct-Input-Gigawords				X	X			
	53	Acct-Output-Gigawords				X	X			
	55	Event-Timestamp			X	X	X			
2865	60	CHAP-Challenge	X							

2868	64	Tunnel-Type		X	X	X	X			
	65	Tunnel-Medium-Type		X						
	66	Tunnel-Client-Endpoint		X	X	X	X			
	67	Tunnel-Server-Endpoint		X	X	X	X			
	69	Tunnel-Password		X						
	81	Tunnel-Private-Group-ID		X	X	X	X			
	82	Tunnel-Assignment-ID		X	X	X	X			
	83	Tunnel-Preference		X					X	

2869	85	Acct-Interim-Interval	X						X	
	87	NAS-Port-Id	X		X	X	X			
	88	Framed-Pool		X						
2868	90	Tunnel-Client-Auth-ID		X	X	X	X			
	91	Tunnel-Server-Auth-ID		X	X	X	X			
3162	97	Framed-IPv6-Prefix		X	X	X	X			
	100	Framed-IPv6-Pool		X						
3576	101	Error-Cause								
4818	123	Delegated-IPv6-Prefix		X	X	X	X			

6911	171	Delegated-IPv6-Prefix-Pool		X						

4679

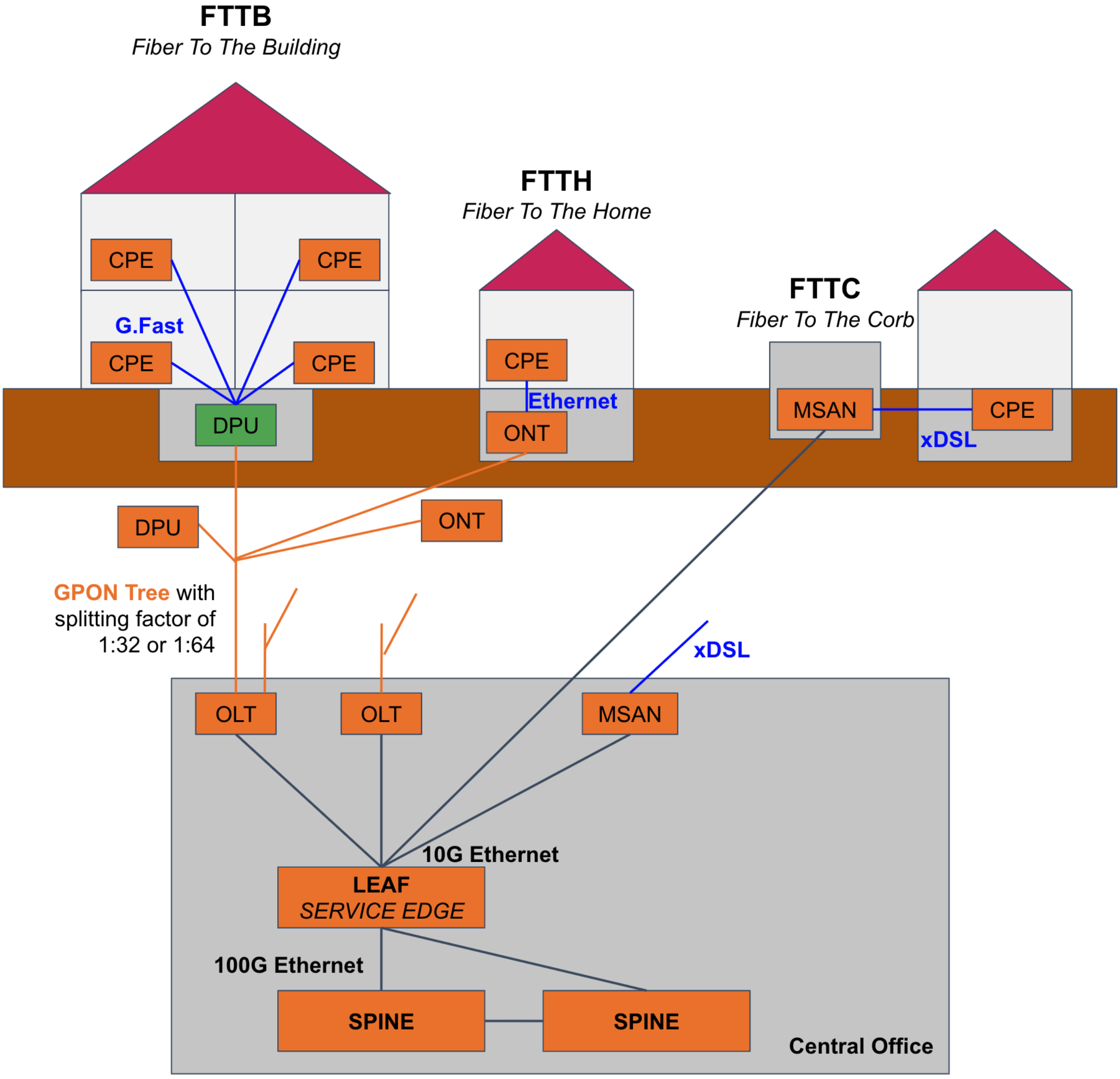
	m-Low-Power								
26-3561-139	Maximum-Interleaving-Delay-Upstream	X	X	X	X	X			
26-3561-140	Actual-Interleaving-Delay-Upstream	X	X	X	X	X			
26-3561-141	Maximum-Interleaving-Delay-Downstream	X	X	X	X	X			
26-3561-142	Actual-Interleaving-Delay-Downstream	X	X	X	X	X			
26-3561-144	Access-Loop-Encapsulation	X	X	X	X	X			

RtBrick	

26-50058-140	RtBrick-LI-Action		X					X	
26-50058-141	RtBrick-LI-Identifier		X					X	
26-50058-142	RtBrick-LI-Direction		X					X	
26-50058-143	RtBrick-LI-MED-Instance		X					X	
26-50058-144	RtBrick-LI-MED-IP		X					X	
26-50058-145	RtBrick-LI-MED-Port		X					X	

4.2.11. Appendix C - Access Deployments

The following picture shows different deployments typically used in access networks.



The LEAF and SPINE are bare metal switches running RBFS where the LEAF is responsible for service edge functionalities like PPPoE subscriber termination and the SPINE builds the actual fabric core. Multiple LEAF switches will be connected to a pair of spine switches.

4.3. RBFS Carrier-Grade Network Address Translation

4.3.1. RBFS Carrier-Grade Network Address Translation Overview

Carrier-Grade NAT

The rise of technologies and the rapid growth of mobile devices and cloud services worldwide resulted in the exhaustion of IPv4 addresses. Broadband service providers face a growing challenge as this 32-bit address is insufficient to meet the ever-growing demand and the cost of obtaining public IP addresses for every individual subscriber. Though the launch of IPv6 can address the IPv4 depletion problem, migrating to an IPv6 network becomes vastly complex and costly. IPv6 also does not provide backward compatibility with IPv4. Consequently, migrating to IPv6 is not an ideal solution for many broadband service providers.

RBFS Carrier-Grade Network Address Translation (CGNAT) is a prominent technology solution that addresses the IPv4 exhaustion challenge of service providers. The solution helps to use private network addresses and to limit the use of publicly routable IPv4 addresses significantly.

RBFS CGNAT allows service providers to serve a large number of their subscribers using a limited number of public IPv4 addresses. It saves costs on public IPv4 addresses and conserves their IPv4 address pools.

Static NAT

Static NAT associates one private IP address with one public IP address. This is a one-to-one mapping and you must manually define the mapping between a private and public address.

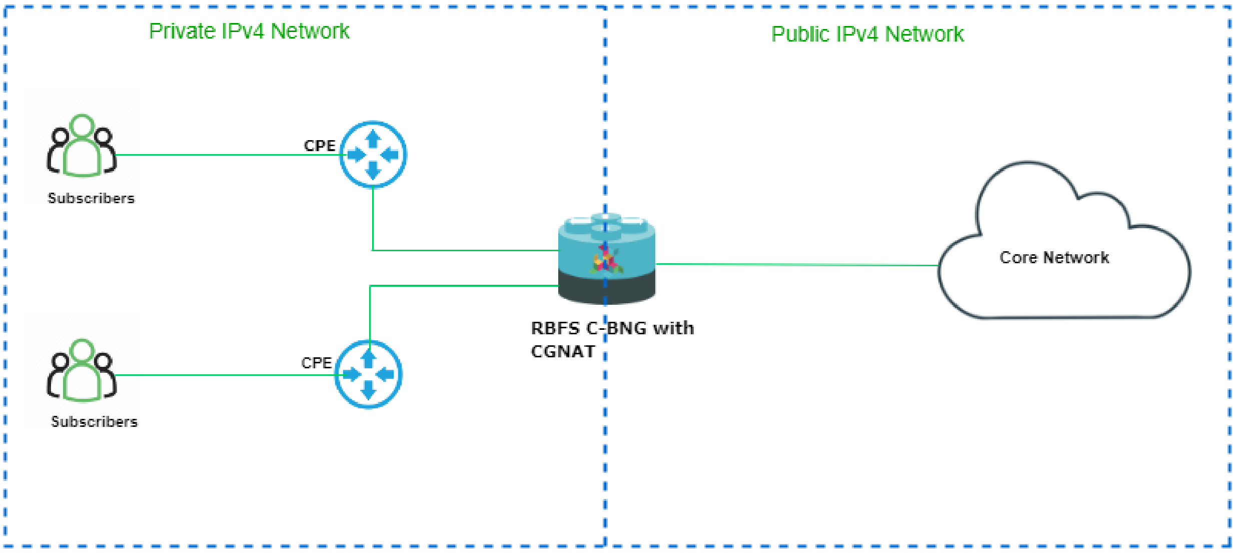
Dynamic NAT

In Dynamic NAT, internal private IPv4 addresses are mapped with a public IPv4 address and this mapping of a private IPv4 address to a public IPv4 address happens dynamically. The router dynamically picks an address from the address pool that is not currently assigned.

NAT444

RBFS CGNAT supports NAT444. NAT 444 refers to three sets of IPv4 addresses: Customer private, ISP private, and public Internet. Network address translation occurs from customer private to ISP private, and then ISP private to public. The NAT444, also known as CGNAT, functionality maps IPv4 subscriber private addresses to IPv4 public addresses.

The following diagram illustrates a high-level view of RBFS CGNAT.



RBFS CGNAT Benefits

RBFS CGNAT allows service providers to provide services to a large number of their subscribers using a limited number of public IPv4 addresses. The benefits include:

Conserve Public IPv4 Addresses

The solution helps service providers conserve expensive public IPv4 addresses by enabling multiple subscribers to share a single public IPv4 address. With CGNAT, one public IPv4 address can manage hundreds of devices within the private network. It reduces the requirement of continually buying additional public IP addresses.

Enhance Security

In addition to stretching the limited pool of public IPv4 addresses even further, CGNAT also provides significant security benefits. It helps to enhance network security by keeping the internal addressing private from the external network. It makes it difficult for attackers to target specific devices on the network by preventing attacks that target specific IP addresses.

Performance at Carrier-Grade

RBFS CGNAT solution implements the functions of IPv4 address translation in the network of service providers. Service providers can deploy NAT in a way that allows multiple subscribers to share a single global IPv4 address and scale to several thousands of address translations. CGNAT operations are implemented at the forwarding plane (hardware level) to ensure optimal performance. This hardware-level implementation achieves high throughput with no performance impact and packet processing occurs without overburdening the CPU. The solution offers carrier-grade scalability with fast translation rates and a large number of IPv4 address and port number translations.

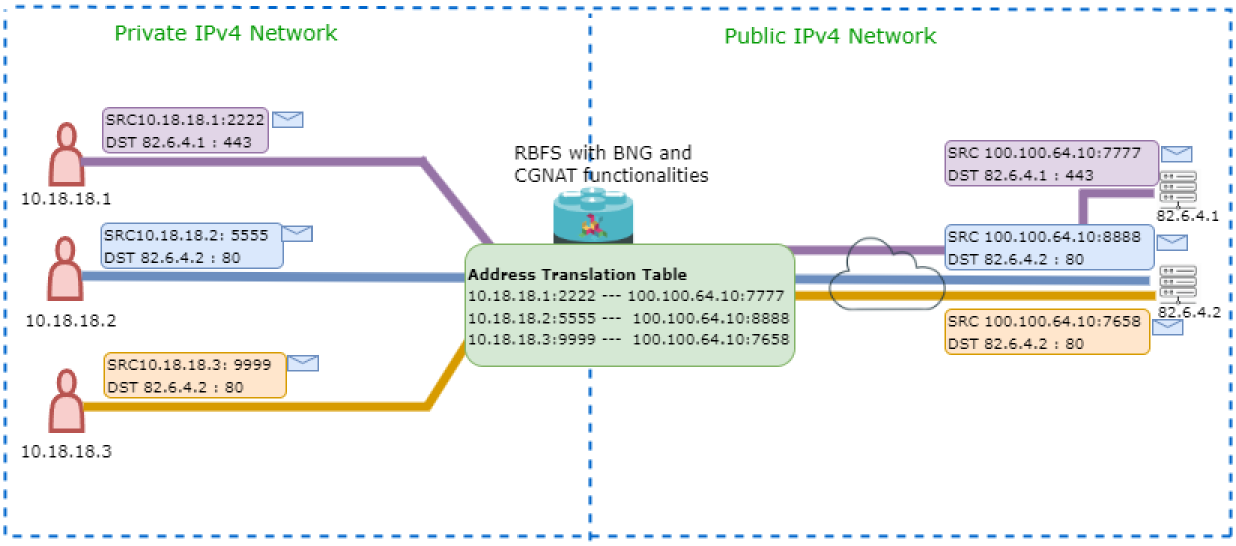
Understanding RBFS CGNAT Implementation

RBFS CGNAT solution has been designed to support Port Address Translation, also known as Network Address Port Translation (NAPT), which has the greatest potential to conserve IPv4 addresses for service providers. NAPT is known as an effective method to allow multiple devices to connect to the Internet using a single public IPv4 address.

Network Address Port Translation

Network Address Port Translation is a dynamic NAT in which port numbers along with IP address are used to identify which traffic belongs to which private IP address. RBFS CGNAT translates the source private IPv4 address and port number to a public source IPv4 address and unique port number. This allows multiple devices with different private IP addresses to use a single public IPv4 address. The unique port number ensures that the traffic is delivered to the correct device.

The following diagrams illustrate how RBFS CGNAT works at a high level.



The diagram shows three subscribers using private IPv4 addresses (10.18.18.1, 10.18.18.2, and 10.18.18.3 send traffic to two different servers (82.6.4.1 and

82.6.4.2) on the public network.

BNG with CGNAT performs the address translation by replacing the source private IPv4 address with the public IPv4 address from the address pool and the source port number with a unique port number. After the address translation, the device forwards the traffic to the destination servers. After the translation, packets have a new source IPv4 address which is the public IPv4 and a unique port number as per the mapping in the address translation table.

CGNAT maintains a translation table with the entries. Entries are records of mapped private IPv4 addresses and port numbers with the public IPv4 addresses and port numbers.

The downstream traffic from the servers traverses to the RBFS CGNAT device. The packets coming from external hosts include the destination address as the (translated) public IPv4 address. CGNAT device performs reversal address translation for these packets as per the translation table mapping for the downstream traffic.

NAPT helps to significantly reduce the number of logs generated as it generates logs only during the allocation and release of each block of ports.

Deterministic NAT

RBFS CGNAT offers support for deterministic NAT mode, which provides a consistent mapping of private IPv4 addresses with public IPv4 addresses and port ranges. This mode ensures a one-to-one mapping of private IPv4 addresses with public IPv4 addresses, allowing you to specify the private address and its matching public address and port range. The given private IPv4 address is always translated to the same public address.

In addition, deterministic NAT guarantees a predetermined and fixed translation for a given internal address with a public IPv4 and port combination, ensuring consistency and stability in the mapping. This feature is particularly useful for applications that involve security protocols, as well as for service providers who need to track subscriber sessions.

Furthermore, deterministic NAT significantly reduces address translation logs, as private IPv4 addresses are always mapped to public IPv4 addresses and port ranges.

Address Translation Table

Port mapping is a feature that allows multiple devices to share the same public IPv4 address with different port numbers. CGNAT generates a unique port number for each subscriber session. Port mapping helps to determine the correct host among the many devices in the private network that use the same public IPv4 address.

RBFS CGNAT maintains an address translation table that maps private IPv4 addresses and port numbers to their corresponding public IPv4 addresses and ports. Whenever an incoming packet arrives, CGNAT checks the translation table to see if a translation entry exists for that packet. If there’s an entry, the CGNAT replaces the source IPv4 address and port number in the packet header with the mapped public IPv4 address and port number.

When a device on the Internet sends packets downstream, the CGNAT software uses the translation table for address translation reversal. It replaces the destination public IPv4 address and port number in the packet header with the corresponding private IPv4 address and port number.

Port Block Allocation

RBFS CGNAT allows Port block allocation (PBA) mode which is an address translation option. Port block size determines the number of ports allocated in a port set. Port blocks have a fixed size that include 64, 128, 256, 512, 1024, and 2048.

A total number of 64512 ports are available for use for a group of subscribers with the same public IPv4 address. Based on the port block size defined in the profile, the number of ports are allocated to each public IPv4 address in the pool. For example, if the block size is defined as 256, 252 ports will be available for subscribers who use the same IPv4 address.

The following table shows the port block size and available ports for a single public IPv4 address in an IPv4 pool for that block size.

Port Block Size	Subscribers per IPv4 address
64	1008
128	504
256	252

Port Block Size	Subscribers per IPv4 address
512	126
1024	63
2048	31

Port Block Size allocation determines how many ports are assigned to individual subscribers who share the same public IPv4 address. The assigned ports are dynamically allocated to the subscribers as required. Whenever a subscriber initiates a connection, an available port from the designated block is assigned to that subscriber.

The Port Block Size that is allocated to each subscriber is determined by various factors, such as the number of subscribers sharing a single public IPv4 address and the expected volume of concurrent connections.

NAT IP Pools and Chaining

A NAT pool contains a range of multiple public IP addresses. You can create multiple pools and associate them with a NAT service. Pool chaining is a method in which you can associate one pool with another. For a pool, you can define 'next pool name'; so that when the pool gets exhausted with the IPv4 addresses, the next pool that is defined will take over.

Aging

Aging refers to the time that a translation entry exists or remains in the address translation table after it was last used.

The entries in the translation table have a finite lifespan as the software implements mechanisms to handle session timeouts.

When a host sends a packet to a destination, CGNAT translates the private IPv4 address and port to a public IPv4 address and port. This mapping is recorded as an entry in the address translation table. The software always looks up the translation table whenever it receives a packet to verify that any entry exists for the packet in the translation table. The software performs the address translation based on the recorded entries and its associated mappings for both inbound and outbound packets.

If there is no activity related to a specific mapping (for example, when there are no incoming or outgoing packets), the mapping will eventually be removed from the address translation table, once idle or unused entries are detected, entries removed from the address translation table to free up resources for the new upcoming traffic flows.

For TCP, idle or unused flows are typically detected by the receipt of TCP FIN-ACK packets, which indicate that both sides of the connection have finished sending data. Once detected, these flows are removed after a configurable aging timer expires.

For UDP, as it is a connectionless protocol, idle or unused flows are detected by periodically polling the flow traffic. As the number of UDP traffic flows increases, the idle flow detection mechanism also increases, ensuring that idle flows are removed in a timely manner. The aging of UDP flows is proportional to the number of traffic flows.

Logging

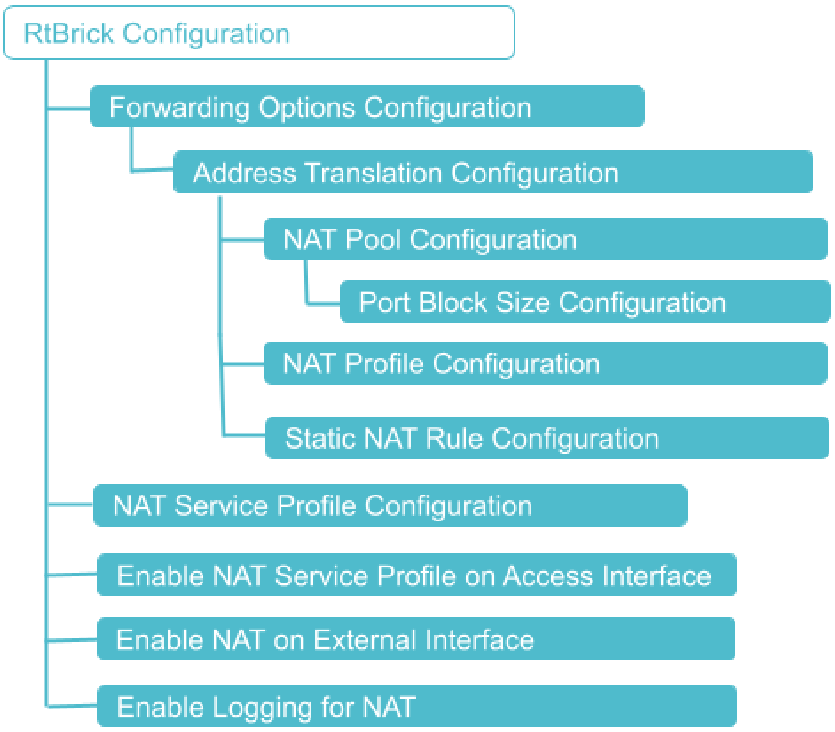
RBFS CGNAT provides traffic monitoring capabilities to track and log network activities. The solution offers a flexible logging mechanism that can store information such as port numbers, time, destination, and address translation details. You can enable logging for CGNAT operations.

Supported Platforms

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

4.3.2. Carrier-Grade NAT Configuration

Configuration Hierarchy


RtBrick Configuration
Forwarding Options Configuration
Address Translation Configuration
NAT Pool Configuration
Port Block Size Configuration
NAT Profile Configuration
Static NAT Rule Configuration
NAT Service Profile Configuration
Enable NAT Service Profile on Access Interface
Enable NAT on External Interface
Enable Logging for NAT

CGNAT Configuration

You must perform the following tasks to configure CG NAT.

1. Configure NAT Pool
2. Configure NAT Port Block Size
3. Configure NAT Profile
4. Configure NAT Rule
5. NAT Service Profile Configuration
6. Enable NAT Service Profile on Access Interface
7. Enable NAT on External Interface
8. Enable Logging for NAT

Configuration Syntax and Commands

The following sections describe the CGNAT configuration syntax and commands.

Configuring NAT Profile

A NAT profile defines how the NAT device has to perform the IPv4 address translation. NAT profile allows you to define an instance, IPv4 address pools, maximum number of translations, port block size and mapping a particular internal IPv4 address with a particular external IPv4 address for a deterministic address translation.

You can create NAT profile for an RBFS instance using the 'instance' option. Also, you can define the TCP or UDP traffic type for the profile.

Syntax:

set forwarding-options address-translation profile <profile-name> <attribute> <value>

Attribute	Description
<profile-name>	Specify the NAT profile name.
deterministic [true]	Specify deterministic as true to enable deterministic NAT for the profile. Deterministic NAT allows subscribers always to connect with a single public IP.
instance	Specify the RBFS instance.
ip-protocol	Specify the protocol: TCP or UDP.
ip-protocol ageing-timeout <ageing-timeout>	Specify the aging time value for the protocol.
max-rules	Specify the maximum number of rules for address translations for a public IPv4 address and for an interface.
pool	Specify the name of the public IP address pool.

The following commands configure the NAT profile named nat_profile1. The nat profile nat_profile1 is configured on the instanced vrf1 with a pool attached nataddr_pool1. Maximum rules are configured as 100 rules and the aging period is configured as 600 seconds for TCP traffic and 300 seconds for UDP traffic.

```
set forwarding-options address-translation profile nat_profile1
set forwarding-options address-translation profile nat_profile1 instance vrf1
set forwarding-options address-translation profile nat_profile1 pool nataddr_pool1
set forwarding-options address-translation profile nat_profile1 max-rules 100
set forwarding-options address-translation profile nat_profile1 ip-protocol TCP
ageing-timeout 600
set forwarding-options address-translation profile nat_profile1 ip-protocol UDP
ageing-timeout 300
set forwarding-options address-translation profile nat_profile1 ip-protocol ALL
```

```
ageing-timeout 300
```

Example Configuration:

```
supervisor@rtbrick>cbng1.rtbrick.net: cfg> show config forwarding-options address-
translation profile
{
  "rtbrick-config:profile": [
    {
      "profile": "nat_profile1",
      "instance": "vrf1",
      "pool": "nataddr_pool1",
      "max-rules": "100",
      "ip-protocol": {
        "TCP": {
          "ageing-timeout": 600
        },
        "UDP": {
          "ageing-timeout": 300
        },
        "ALL": {
          "ageing-timeout": 300
        }
      }
    }
  ]
}
```

NAT Pool Configuration

A NAT IP address pool includes a set of public IPv4 addresses that are used for network address translation. You can create multiple public IPv4 address pools and one pool includes a range of public IPv4 addresses. These pools allocate public IPv4 addresses to subscribers during address translation. While configuring a pool, you can define the group of public IPv4 addresses belonging to that pool by specifying the lowest and highest IP addresses.

The system allows you to create multiple pools and define the association among them. You can define the 'next-pool-name' that takes over when the current pool gets exhausted with the IPv4 addresses. When one pool gets exhausted, the next pool takes over and starts serving the IP addresses to subscribers when the address translation occurs.

In addition, you can define the port block allocation by specifying the port block size for that pool. So, each public IPv4 in the pool can be allocated a certain number of ports based on the port block size defined.

Syntax:

set forwarding-options address-translation pool <pool-name> <attribute> <value>

Attribute	Description
<pool-name>	Specify the name of the address pool.
ipv4-address	Specify both the highest and lowest IPv4 addresses in the range of IPv4 addresses for the pool.
ipv4-address high	Specify the highest IPv4 address in the address pool. You must specify the highest IP address in the range of IP addresses.
ipv4-address low	Specify the lowest IPv4 address in the address pool. You must specify the lowest IP address in the range of IP addresses.
next-pool-name	Specify the name of the next address pool that is to be used when the current address pool is allocated completely.
port-block-size	The number of ports allocated in a block. The default value is 256. For information, about port block allocation, see the section "Port Block Allocation".

Example Configuration:

The following commands configure nataddr_pool1 as the NAT pool and nataddr_pool2 as the next pool, and the port block size as defined 1024.

It indicates NAT pool nataddr_pool1 contains a range of public IPv4 addresses from 100.100.100.1 to 100.100.100.5 with the port block size 1024. With the port block size 1024, the system can allocate 63 ports to subscribers who have the same IPv4 address.

When the pool nataddr_pool1 has fully allocated its IPv4 addresses, the next pool named nataddr_pool2 will start allocating IPv4 addresses from its pool. The pool nataddr_pool2 includes a rage of IPv4 address from 100.100.101.1 to 100.100.101.150.

```
set forwarding-options address-translation pool nataddr_pool1
set forwarding-options address-translation pool nataddr_pool1 next-pool-name
nataddr_pool2
set forwarding-options address-translation pool nataddr_pool1 port-block-size 1024
set forwarding-options address-translation pool nataddr_pool1 ipv4-address low
100.100.100.1
set forwarding-options address-translation pool nataddr_pool1 ipv4-address high
100.100.100.5
set forwarding-options address-translation pool nataddr_pool2 ipv4-address low
100.100.101.1
set forwarding-options address-translation pool nataddr_pool2 ipv4-address high
100.100.101.150
```

Example Configuration:

```
supervisor@rtbrick: cfg> show config forwarding-options address-translation pool
{
  "rtbrick-config:pool": [
    {
      "pool-name": "nataddr_pool1",
      "next-pool-name": "nataddr_pool2",
      "port-block-size": "1024",
      "ipv4-address": {
        "low": "100.100.100.1",
        "high": "100.100.100.5"
      }
    },
    {
      "pool-name": "nataddr_pool2",
      "port-block-size": "128",
      "ipv4-address": {
        "low": "100.100.101.1",
        "high": "100.100.101.150"
      }
    }
  ]
}
```

NAT Rule Configuration

You can define NAT rules only for static NAT. A NAT rule defines a match condition and a corresponding action. After you specify NAT rules, each packet is matched with each NAT rule. If a packet matches the condition specified in a rule, then the action corresponding to that match occurs. Match rules govern how the translation of private IPv4 addresses to public IPv4 addresses is performed.

With NAT rules, you can define how address translation is applied to traffic, and how to handle various protocols and data traffic, such as TCP and UDP, to ensure proper address translation and the mappings of private addresses to public addresses.

Rules also define how to handle inbound and outbound traffic, different protocols, and data traffic such as TCP and UDP for ensuring the proper address translation of traffic.

Syntax:

set forwarding-options address-translation rule <rule-name> <attribute> <value>

Attribute	Description
<rule-name>	Specify the name of the rule.
ordinal <ordinal-value>	Specify the ordinal value. An ordinal value is a numerical representation that indicates its relative position or order.
ordinal <ordinal-value> instance	Specify the RBFS instance name.
ordinal <ordinal-value> ip-protocol [tcp/udp]	Specify the IP protocol, TCP or UDP.
ordinal <ordinal-value> local [ipv4-address/port]	Specify the private IPv4 address or port number that needs to be translated.
ordinal <ordinal-value> public [ipv4-address/port]	Specify the public IPv4 address. This public IP will be mapped with the private IP in the translation table.

Port Block Size Configuration

You can configure port block size for an IP address pool. Based on the block size set, the number of ports is allocated to per IPv4 address and per protocol (TCP or UDP).

Syntax:

set forwarding-options address-translation pool <pool-name> **port-block-size** <value>

Attribute	Description
<pool-name>	Specify the name of the pool.
port-block-size	Specify the value. Supported values include 64, 128, 256, 512, 1024, and 2048.

The following commands configure the public IP pool nataddr_pool10 and port-block-size as 2048. The address pool contains a range of public IPv4 addresses from 100.100.102.51 to 100.100.102.100. With the port block size 2048, it can allocate 31 ports to each IP address in the pool.

```
set forwarding-options address-translation pool nataddr_pool10
set forwarding-options address-translation pool nataddr_pool10 port-block-size
2048
set forwarding-options address-translation pool nataddr_pool10 ipv4-address low
100.100.102.51
set forwarding-options address-translation pool nataddr_pool10 ipv4-address high
100.100.102.100
```

Example:

```
supervisor@rtbrick: cfg> show config forwarding-options address-translation pool
nataddr_pool10
{
  "rtbrick-config:pool": [
    {
      "pool-name": "nataddr_pool10",
      "port-block-size": "2048",
      "ipv4-address": {
        "low": "100.100.102.51",
        "high": "100.100.102.100"
      }
    }
  ]
}
```

Configuring NAT Service Profile

You must create a NAT service profile and attach the service profile with the access interface for enabling CGNAT on the interface.

Syntax:

set access service-profile <profile-name> <attribute> <value>

Attribute	Description
<profile-name>	Name of the service profile.
profile <profile>	Specify the profile name for the address translation.

Enable NAT Service Profile on the Access Interface

It is required to attach the NAT service profile to the access interface for enabling address translation on the interface.

Syntax:

set access interface [double-tagged | single-tagged | untagged] <interface-name> <outer-vlan-min> <outer-vlan-max> <inner-vlan-min> <inner-vlan-max> **service-profile-name** <service-profile-name>

Attribute	Description
service-profile	Configure global service profile.
<outer-vlan-min>	Specify the minimum number of outer VLANs. Allowed range: 1 - 4094.
<outer-vlan-max>	Specify the maximum number of outer VLANs. Allowed range 1 - 4094.
<inner-vlan-min>	Specify the minimum number of inner VLANs. Allowed range: 1 - 4094.
<inner-vlan-max>	Specify the maximum number of inner VLANs. Allowed range 1 - 4094.
service-profile-name	Specify the name of the service profile.

The following commands attach service profile nat_service to the interface (double-tagged) ifp-0/0/17. The configuration shows the minimum number of outer VLANs as 1000, the maximum number of outer VLANs as 1007, the minimum number of inner VLANs as 84 and the maximum number of inner VLANs as 4084. The access type configured is IPoE, service profile is NAT service. AAA profile name is ipoe-aaa and the gateway IFL is lo-0/0/0/100.

IPoE subscriber with outer VLAN between 1000 and 1007, and inner VLAN between 84 and 4084 will be matched with this NAT service profile and a corresponding action will be taken. Anything outside these vlans will not have any action from this NAT service profile.

```
set access interface double-tagged ifp-0/0/17 1000 1007 84 4084
set access interface double-tagged ifp-0/0/17 1000 1007 84 4084 access-type IPoE
set access interface double-tagged ifp-0/0/17 1000 1007 84 4084 access-profile-
name ipoe
set access interface double-tagged ifp-0/0/17 1000 1007 84 4084 service-profile-
name nat_service
set access interface double-tagged ifp-0/0/17 1000 1007 84 4084 aaa-profile-name
ipoe-aaa
set access interface double-tagged ifp-0/0/17 1000 1007 84 4084 gateway-ifl lo-
0/0/0/100
```

```
{
  "rtbrick-config:interface": {
    "double-tagged": [
      {
        "interface-name": "ifp-0/0/17",
        "outer-vlan-min": 1000,
        "outer-vlan-max": 1007,
        "inner-vlan-min": 84,
        "inner-vlan-max": 4084,
        "access-type": "IPoE",
        "access-profile-name": "ipoe",
        "service-profile-name": "nat_service",
        "aaa-profile-name": "ipoe-aaa",
        "gateway-ifl": "lo-0/0/0/100"
      }
    ]
  }
}
```

The following commands configure a double-tagged interface ifp-0/0/6 and outer VLAN minimum value is 1000, maximum value as 1007, inner VLAN minimum value as 84 and maximum value as 4084. The access type configured is PPPoE, service profile is NAT service. AAA profile name is configured as ipoe-aaa.

It indicates PPPoE subscriber with outer VLAN between 1000 and 1007, and inner VLAN between 84 and 4084 will be matched with this NAT service profile and a corresponding action will be taken. Anything outside these vlans will not have any action from this NAT service profile.

```
set access interface double-tagged ifp-0/0/16 1000 1007 84 4084
set access interface double-tagged ifp-0/0/16 1000 1007 84 4084 access-type PPPoE
set access interface double-tagged ifp-0/0/16 1000 1007 84 4084 access-profile-
name pppoe
set access interface double-tagged ifp-0/0/16 1000 1007 84 4084 service-profile-
name nat_service
set access interface double-tagged ifp-0/0/16 1000 1007 84 4084 aaa-profile-name
ipoe-aaa
```

Example:

```
{
  "rtbrick-config:interface": {
    "double-tagged": [
      {
        "interface-name": "ifp-0/0/16",
        "outer-vlan-min": 1000,
        "outer-vlan-max": 1007,
        "inner-vlan-min": 84,
        "inner-vlan-max": 4084,
        "access-type": "PPPoE",
        "access-profile-name": "pppoe",
        "service-profile-name": "nat_service",
        "aaa-profile-name": "ipoe-aaa"
      },
    ]
  }
}
```

Enable NAT on External Interface

It is required to enable the NAT on the external (core-facing) interface.

Syntax:

set interface <interface-name> **unit** <unit-id> **address-translation direction** <public>

Attribute	Description
<interface-name>	Name of the interface.
<unit-id>	Configure the number of sub-interfaces under the physical interface.
public	Specify 'public' for the external interface.

The following commands configure the external interface ifp-0/1/64 for IPv4 address translation. 'Unit' logical identifier for this physical interface. Direction 'public' shows the configuration on external interface for address translation.

```
set interface ifp-0/1/64 unit 100
set interface ifp-0/1/64 unit 100 address-translation
set interface ifp-0/1/64 unit 100 address-translation direction public
```

```
supervisor@rtbrick.net: cfg> show config interface ifp-0/1/64 unit 100
{
  "rtbrick-config:unit": [
    {
      "unit-id": 100,
      "address-translation": {
        "direction": "public"
      }
    }
  ]
}
```

Enable Logging for NAT

You can optionally enable logging for CGNAT operations.

> All RBFS logs and related information is available in the *RBFS Logging User Guide*. For the list of RBFS logs, see Log Reference.

set log bd <name> <options>

Attribute	Description
level	Specify the log level.
module	Specify the log module.
plugin-alias	Specify the plugin-alias URL. Plugin-alias is an external logging host server to which you can export logs. For example, Graylog.

The following commands configure logging for NAT module natd with a log level 'debug'.

```
set log bd natd
set log bd natd module nat
set log bd natd module nat level debug
```

```
supervisor@rtbrick.net: cfg> show config log bd natd
{
  "rtbrick-config:bd": [
    {
      "bd-name": "natd",
      "module": [
        {
          "module-name": "nat",
          "level": "debug"
        }
      ]
    }
  ]
}
```

4.3.3. Carrier Grade NAT Operational Commands

Carrier Grade NAT Show Commands

The RBFS CGNAT operational commands provide detailed information about the address translation operations.

NAT Pool Information

This command displays information about IPv4 address pool for a user.

Syntax:

show address-translation allocation <options>

Option	Description
pool	Specify the pool name.
user	Specify the user name.

Example for NAT pool allocation.

```
supervisor@rtbrick.net: cfg> show address-translation allocation pool
Pool: pool1
  Profile: nat_profile1
  Instance: ip2vrf
  Allocated: 100.00%
Pool: pool2
  Profile: nat_profile1
  Instance: ip2vrf
  Allocated: 100.00%
supervisor@rtbric
```

Example for allocation details for subscribers

```
supervisor@rtbrick.net: cfg> show address-translation allocation user
User                             Original Address     Translated Address   Port
Range
ppp-0/1/20/72339069014638594     10.100.128.1         100.100.100.100      14912 -
14975
ppp-0/1/20/72339069014638594     10.100.128.1         100.100.100.100      18944 -
19007
ppp-0/1/20/72339069014638595     10.100.128.3         100.100.100.101      14976 -
15039
ppp-0/1/20/72339069014638595     10.100.128.3         100.100.100.101      18880 -
18943
ppp-0/1/20/72339069014638596     10.100.128.5         100.100.100.101      15104 -
15167
ppp-0/1/20/72339069014638596     10.100.128.5         100.100.100.101      17856 -
17919
ppp-0/1/20/72339069014638597     10.100.128.7         100.100.100.100      15104 -
15167
ppp-0/1/20/72339069014638597     10.100.128.7         100.100.100.100      17792 -
17855
ppp-0/1/20/72339069014638598     10.100.128.9         100.100.100.101      15040 -
15103
ppp-0/1/20/72339069014638598     10.100.128.9         100.100.100.101      17792 -
17855
ppp-0/1/20/72339069014638599     10.100.128.11        100.100.100.100      15040 -
```

```
15103
```

NAT Rule Details

Syntax:

show address-translation rule <options>

Option	Description
-	Without any option, the command displays the information for all rules.
instance	Displays NAT rule information for the specified instance.
summary	Displays summary of the information for the NAT rule.
user	Displays user information for the NAT rule.

Example for address translation rule for an instance.

```
supervisor@rtbrick.net: cfg> show address-translation rule instance default
Instance: default
User                             Protocol   Original Address        Translated
Address      Direction
ipoe-0/0/18/216454257090494481   tcp        11.100.128.24, 60011    100.100.100.1,
1048     Both
ppp-0/0/17/72339069014638604     tcp        11.100.128.25, 60011    100.100.100.2,
1034     Both
ipoe-0/0/18/216454257090494481   tcp        11.100.128.24, 60012    100.100.100.1,
1049     Both
ppp-0/0/17/72339069014638604     tcp        11.100.128.25, 60012    100.100.100.2,
1035     Both
ipoe-0/0/18/216454257090494481   tcp        11.100.128.24, 60013    100.100.100.1,
1050     Both
ppp-0/0/17/72339069014638604     tcp        11.100.128.25, 60013    100.100.100.2,
1036     Both
ipoe-0/0/18/216454257090494481   tcp        11.100.128.24, 60014    100.100.100.1,
1051     Both
ppp-0/0/17/72339069014638604     tcp        11.100.128.25, 60014    100.100.100.2,
1037     Both
ipoe-0/0/18/216454257090494481   tcp        11.100.128.24, 60015    100.100.100.1,
1052     Both
ppp-0/0/17/72339069014638604     tcp        11.100.128.25, 60015    100.100.100.2,
1038     Both
ipoe-0/0/18/216454257090494481   tcp        11.100.128.24, 60016    100.100.100.1,
1053     Both
ppp-0/0/17/72339069014638604     tcp        11.100.128.25, 60016    100.100.100.2,
1039     Both
ipoe-0/0/18/216454257090494481   tcp        11.100.128.24, 60017    100.100.100.1,
1054     Both
ppp-0/0/17/72339069014638604     tcp        11.100.128.25, 60017    100.100.100.2,
1040     Both
ipoe-0/0/18/216454257090494481   tcp        11.100.128.24, 60018    100.100.100.1,
1024     Both
ppp-0/0/17/72339069014638604     tcp        11.100.128.25, 60018    100.100.100.2,
1041     Both
ipoe-0/0/18/216454257090494481   tcp        11.100.128.24, 60019    100.100.100.1,
1025     Both
ppp-0/0/17/72339069014638604     tcp        11.100.128.25, 60019    100.100.100.2,
1042     Both
ipoe-0/0/18/216454257090494481   tcp        11.100.128.24, 60020    100.100.100.1,
1026     Both
```

Example for address translation rule summary. The example summary shows the total number of NAT rules for TCP flows.

```
supervisor@rtbrick.net: cfg> show address-translation rule summary
Instance: default
  Source     Protocol      Rules
  Dynamic    tcp              62
  Total Dynamic Rules         62
```

Example for address translation rules for subscribers.

```
supervisor@rtbrick: cfg> show address-translation rule
Instance: ip2vrf
User                             Protocol   Original Address        Translated
Address      Direction
ipoe-0/1/21/216454257090494746   tcp        11.100.129.0, 65001
100.100.100.101, 36874  Both
ipoe-0/1/21/216454257090494975   tcp        11.100.130.0, 65001
100.100.100.101, 60170  Both
ipoe-0/1/21/216454257090495232   tcp        11.100.131.0, 65001
100.100.101.101, 38730  Both
ppp-0/1/20/72339069014638594     tcp        10.100.128.1, 65001
100.100.100.100, 14914  Both
ipoe-0/1/21/216454257090494501   tcp        11.100.128.1, 65001
100.100.100.100, 2122   Both
ppp-0/1/20/72339069014638722     tcp        10.100.129.1, 65001
100.100.100.100, 19924  Both
ipoe-0/1/21/216454257090494747   tcp        11.100.129.1, 65001
100.100.100.100, 36746  Both
```

NAT Trap Statistics

Syntax:

show address-translation trap statistics

This command displays the information of packets trapped in the NAT daemon.

Example:

```
supervisor@rtbrick: cfg> show address-translation trap statistics
Protocol   Flags                     Status             Count
tcp        -|syn|-|-|-|-|-|-|-       Success           152958
tcp        -|syn|-|-|-|-|-|-|-       Failure            23186
Total Successful Traps:                                152958
Total Failed Traps:                                     23186
```

NAT Error Details

This command displays NAT error information.

Syntax:

show address-translation error

Example for address translation platform resource information.

```
supervisor@rtbrick.net: cfg> show address-translation error
INSTANCE - Packet Table
    Counter        : 4
    Current Update : Thu May 02 10:05:52 GMT +0000 2024
    Last Update    : Thu May 02 10:05:52 GMT +0000 2024
POOL_CFG - Pool Flush
    Counter        : 1
    Current Update : Thu May 02 10:05:43 GMT +0000 2024
    Last Update    : Thu May 02 10:05:43 GMT +0000 2024
```

4.4. RBFS HTTP Redirect Service

4.4.1. RBFS HTTP Redirect Service Overview

This document provides information about RBFS HTTP Redirect Service. For enabling HTTP Redirect service, it is required to create, associate, and apply subscriber filters (ACLs) for the subscribers. For information about subscriber filters and how to configure these filters, refer to the RBFS Subscriber Filters User Guide.

RBFS HTTP Redirect service allows network service providers to intercept and redirect HTTP request traffic from subscribers to a designated captive portal instead of the original destination. This service is useful in a multitude of use cases, ranging from subscriber re-authentication to enforcing acceptance of network usage policies.

This captive portal is a webpage where the redirected subscribers are landed up to fulfill certain actions or conditions before they are granted broader access to the network resources. There are various reasons why captive portals can be set up, such as the following:

- Accept the terms of service.
- Receive and manage HTTP requests to unauthorized web resources.
- Present a web page that requires the completion of certain actions from the subscriber.
- Serve commercial communication or network usage policy messages.

By implementing the RBFS HTTP Redirect Service, network service providers can efficiently manage user access and enforce compliance with network regulations and policies, ultimately enhancing the overall security and user experience within their network environment.

Based on the applied filters, RBFS performs three actions which are accept, drop, and redirect. The action Accept allows the subscribers to access the network resource that they request. The Drop action restricts the subscriber from accessing the network resource. Finally, the Redirect action, if enabled HTTP redirect service, takes the subscriber to a different portal where the service provider wants to fulfill certain actions by the subscriber before accessing the network resource.

In addition to RBFS Subscriber Filters, a redirect action is supported by RBFS through the utilization of Ascend Data Filters (ADF), as described in the RBFS RADIUS Services Guide.

The RBFS HTTP Redirect Service together with the Subscriber Filters empowers network service providers to intercept and redirect HTTP request traffic from subscribers, guiding it towards a designated captive portal instead of its original destination.

How RBFS HTTP Redirect Service Works

HTTP requests from subscribers to any destination are intercepted by the RBFS HTTP Redirect service, if the subscriber filter rules are applied to the subscriber. In response to the request from the subscriber to access the network, the HTTP Redirect service provides the HTTP status code 302 along with the new URL to guide the subscriber to the new destination. To make it possible, the RBFS Subscriber Filters are employed, enabling the service to decide which requests should be redirected and which requests should be passed directly without redirection. In addition to RBFS Subscriber Filters, a redirect action is supported by RBFS through the utilization of Ascend Data Filters (ADF), as described in the RBFS RADIUS Services Guide.

Enabling HTTP Redirect Service Using RADIUS ADF

Both the RBFS Subscriber Filters and the HTTP Redirect Service can be dynamically enabled, disabled, and updated through RADIUS access-accept and CoA requests without requiring the re-establishment of the subscriber session. This flexibility allows network administrators to efficiently manage and modify these services as needed without disrupting subscriber connectivity.

RBFS provides a set of vendor-specific attributes to control the RBFS Subscriber Filters and the HTTP Redirect Service. The attributes include:

- VSA 26-50058-75 - RtBrick-HTTP-Redirect-URL
- VSA 26-50058-76 - RtBrick-IPv4-ACL-IN
- VSA 26-50058-77 - RtBrick-IPv4-ACL-OUT
- VSA 26-50058-78 - RtBrick-IPv6-ACL-IN
- VSA 26-50058-79 - RtBrick-IPv6-ACL-OUT

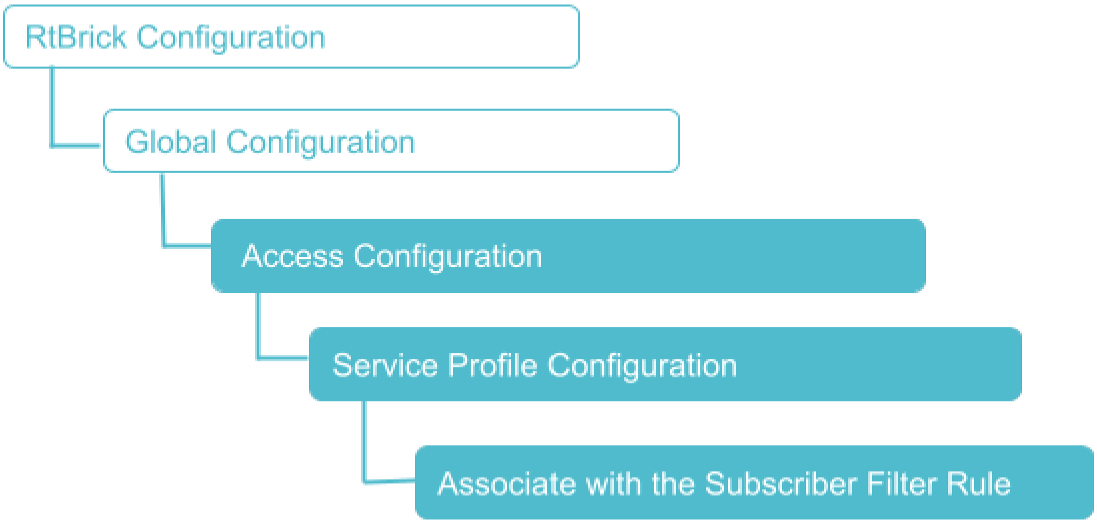
For more information about these attributes, see RBFS RADIUS Services Guide.

Limitations

The HTTP Redirect Service is currently supported only for IPoE subscribers. HTTP Redirect service can redirect only HTTP traffic. Typically, the majority of web requests are in HTTPS format, which cannot be redirected. Due to this limitation, most end-user devices, including PCs, smartphones, and tablets, automatically attempt to access specific well-known URLs to search for captive portals immediately after establishing a network connection. Two common examples are http://connectivitycheck.gstatic.com (vendor-independent) and http://captive.apple.com (Apple devices).

4.4.2. HTTP Redirect Service Configuration

The configuration hierarchy for the HTTP Redirect service is illustrated in the diagram.


RtBrick Configuration
Global Configuration
Access Configuration
Service Profile Configuration
Associate with the Subscriber Filter Rule

Service Profile Configuration

Syntax:

set access service-profile profile-name <profile-name> <attribute> <value>

Attribute	Description
<profile-name>	Service profile name.
<http-redirect>	HTTP redirect service configuration.
<url>	HTTP redirect target URL.
<acl>	Subscriber ACL (filter) configuration.
<ipv4-acl-in>	IPv4 upstream ACL (ingress from subscriber).
<ipv4-acl-out>	IPv4 downstream ACL (egress to subscriber).
<ipv6-acl-in>	IPv6 upstream ACL (ingress from subscriber).
<ipv6-acl-out>	IPv6 downstream ACL (egress to subscriber).

The following example shows a service profile named HTTP, redirect URL as the redirect destination. The 'ipv4-acl-in' ACL is applied as the filter criteria or ACL rule for this redirection.

```
supervisor@router: cfg> show config access service-profile HTTP
{
  "rtbrick-config:service-profile": [
    {
      "profile-name": "HTTP",
      "http-redirect": {
        "url": "https://www.rtbrick.com"
      },
      "acl": {
        "ipv4-acl-in": "redirect-acl-in"
      }
    }
  ]
}
```

4.5. RBFS Subscriber Filters

4.5.1. RBFS Subscriber Filters Overview

RBFS Subscriber Filters, also referred to as subscriber ACLs, consist of a set of rules defining packet match criteria and actions. There are separate rules for IPv4 and IPv6 downstream (egress to subscriber) and upstream (ingress from subscriber) packets. These rules support various match criteria and actions, some of which are specific to address families or directions. Each rule is assigned a priority, and the decision between multiple matching rules is based on these priorities, where lower values take precedence.

The available actions include accept, drop, or http-redirect where the last one refers to the RBFS HTTP Redirect Service. When the action is drop, matching traffic is silently discarded. The filters are categorized into two primary types, namely l3v4 for IPv4 and l3v6 for IPv6, applicable to either ingress or egress direction.

To apply these filters to subscribers, there are two ways. They can be applied through the access service-profile or directly using the corresponding RADIUS attributes with the second method taking priority.

About the Match Criteria

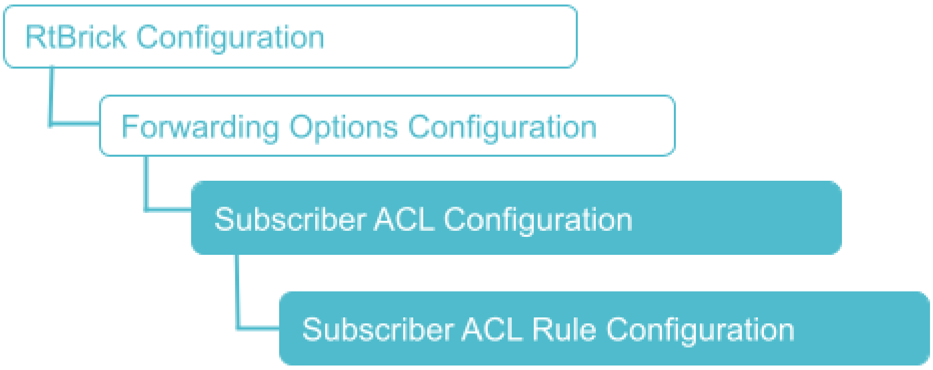
When multiple match criteria are defined within a single rule, they are treated as a logical AND operation, requiring all criteria to be met for the rule to be considered as a match. However, using unsupported match criteria, such as destination-ipv4-subscriber-prefix in ingress (upstream), can potentially lead to session termination. In the case of CoA (Change of Authorization), the filter assignment is rejected using CoA NAK, if such unsupported criteria are encountered.

Even if filters are assigned to a subscriber, those filters are applied globally, indicating that all traffic from all interfaces and subscribers is evaluated against all rules. Consequently, RBFS has introduced specific options to restrict rules to individual subscribers. For ingress (upstream) rules, it is recommended to enable the subscriber-ifl option, ensuring that only traffic received from the corresponding subscriber is matched. With the subscriber-ifl option, packets are matched based on incoming subscriber IFL. However, this option is not supported in egress(downstream), requiring the limitation of traffic using subscriber address prefix information. Thus, RBFS introduced the options source-ipv4-subscriber-prefix, source-ipv6-subscriber-prefix, destination-ipv4-subscriber-prefix, destination-ipv6-subscriber-prefix, source-ipv6-delegated-subscriber-prefix, and destination-ipv6-delegated-subscriber-prefix. With these options enabled, the dynamically assigned subscriber address prefix is automatically integrated into the corresponding filter instance to constrain those rules to a specific subscriber.

Improperly configured filters assigned to one subscriber may create a negative impact on other subscribers as well.

4.5.2. Subscriber Filters Configuration

The configuration hierarchy for Subscriber ACL is illustrated in the diagram.


RtBrick Configuration
Forwarding Options Configuration
Subscriber ACL Configuration
Subscriber ACL Rule Configuration

Configure Subscriber Filters

Syntax:

set forwarding-options subscriber-acl <l3v4|l3v6> **rule** <rule-name> **ordinal** <ordinal-value> <option> <attribute> <value>

Attribute	Description
<l3v4 | l3v6>	Specify l3v4 for IPv4 and l3v6 for IPv6.
<rule-name>	Subscriber ACL rule name.
<ordinal-value>	The mandatory ordinal value is used to differentiate multiple rules within the rule set.
action	The desired action, which can be either permit, drop or http-redirect.
priority <priority-value>	The priority of the rule, where the lower value has a higher priority.
match	The match criteria.

Configuring Subscriber Filter Match Criteria

Syntax:

set forwarding-options subscriber-acl <l3v4|l3v6> **rule** <rule-name> **ordinal** <ordinal-value> **match** <attribute> <value>

Attribute	Description
destination-ipv4-prefix <destination-ipv4-prefix>	Packets are matched when the IPv4 destination address is within the defined prefix.
destination-ipv4-prefix-list <destination-ipv4-prefix-list>	Packets are matched when the IPv4 destination address is within one of the prefixes listed in the defined prefix list.
destination-ipv4-subscriber-prefix <true>	Packets are matched when the IPv4 destination address is within the dynamically assigned subscriber IPv4 address prefix (sometimes, referred to framed prefix). Consequently, this option shares similarity to the destination-ipv4-prefix, where the prefix is automatically set to the dynamic subscriber prefix. This option is allowed only in the egress (downstream) direction.
destination-ipv6-prefix <destination-ipv6-prefix>	Packets are matched when the IPv6 destination address is within the defined prefix.

Attribute	Description
destination-ipv6-prefix-list <destination-ipv6-prefix-list>	Packets are matched when the IPv6 destination address is within one of the prefixes listed in the defined prefix list.
destination-ipv6-subscriber-prefix <true>	Packets are matched when the IPv6 destination address is within the dynamically assigned subscriber IPv6 address prefix (sometimes, referred to framed prefix). Consequently, this option shares similarity to the destination-ipv4-prefix, where the prefix is automatically set to the dynamic subscriber prefix. This option is allowed only in the egress (downstream) direction.
destination-ipv6-delegated-subscriber-prefix <destination-ipv6-delegated-subscriber-prefix>	This option is similar to the destination-ipv6-subscriber-prefix using the dynamically delegated prefix instead.
source-ipv4-prefix <source-ipv4-prefix>	Packets are matched when the IPv4 source address is within the defined prefix.
source-ipv4-prefix-list <source-ipv4-prefix-list>	Packets are matched when the IPv4 source address is within one of the prefixes listed in the defined prefix list.
source-ipv4-subscriber-prefix <source-ipv4-subscriber-prefix>	Packets are matched when the IPv4 source address is within the dynamically assigned subscriber IPv4 address prefix (sometimes, referred to framed prefix). Consequently, this option shares similarity to source-ipv4-prefix, where the prefix is automatically set to the dynamic subscriber prefix. This option is allowed only in the ingress (upstream) direction.
source-ipv6-prefix <source-ipv6-prefix>	Packets are matched when the IPv6 source address is within the defined prefix.
source-ipv6-prefix-list <source-ipv6-prefix-list>	Packets are matched when the IPv6 source address is within one of the prefixes listed in the defined prefix list.

Attribute	Description
source-ipv6-subscriber-prefix <source-ipv6-subscriber-prefix>	Packets are matched when the IPv6 source address is within the dynamically assigned subscriber IPv6 address prefix (sometimes, referred to framed prefix). Consequently, this option shares similarity to source-ipv4-prefix, where the prefix is automatically set to the dynamic subscriber prefix. This option is allowed only in the ingress (upstream) direction only.
source-ipv6-delegated-subscriber-prefix true	This option is similar to the source-ipv6-subscriber-prefix using the dynamically delegated prefix instead. Note: To disable this configuration, use the delete form of the command.
source-l4-port <source-l4-port>	Packets are matched based on the layer 4 source port (TCP or UDP port).
destination-l4-port <destination-l4-port>	Packets are matched based on the layer 4 destination port (TCP or UDP port).
ip-protocol <ip-protocol>	Packets are matched depending on the IP protocol (TCP or UDP). However, this option is not compatible with the MPLS-encapsulated traffic. Consequently, filtering based on IP protocol is not possible for MPLS traffic received from the core. For instance, it is feasible to drop all traffic to port 80, but it is not possible to selectively drop only TCP traffic while permitting UDP traffic when receiving traffic with an MPLS label.
subscriber-ifl <true>	Packets are matched based on incoming subscriber IFL. This option is allowed only in the ingress (upstream) direction.

Configuring Subscriber Filter Actions

Syntax:

set forwarding-options subscriber-acl <l3v4|l3v6> **rule** <rule-name> **ordinal** <ordinal-value> **action** <attribute> <value>

Attribute	Description
permit	Forward packets.
drop	Silently discard packets.
http-redirect true	Specify true to enable the redirect service. Note: To disable http-redirect, use the delete form of the command.

The following example shows an IPv4 subscriber filter configuration in which the applied ACL is ipv4-acl-in.

```
supervisor@rtbrick: cfg> show config forwarding-options subscriber-acl l3v4
{
  "rtbrick-config:l3v4": {
    "rule": [
      {
        "rule-name": "ipv4-acl-in",
        "ordinal": [
          {
            "ordinal-value": 1,
            "match": {
              "destination-l4-port": 80,
              "ip-protocol": "TCP"
            },
            "action": {
              "http-redirect": "true"
            },
            "priority": 1001
          }
        ]
      }
    ]
  }
}
```

Attaching the ACL Rule

Syntax:

set access service-profile profile-name <profile-name> <attribute> <value>

Attribute	Description
<profile-name>	Service profile name.
<http-redirect>	HTTP redirect service configuration.
<url>	HTTP redirect target URL.
<acl>	Subscriber ACL (filter) configuration.

Attribute	Description
<ipv4-acl-in>	IPv4 upstream ACL (ingress from subscriber).
<ipv4-acl-out>	IPv4 downstream ACL (egress to subscriber).
<ipv6-acl-in>	IPv6 upstream ACL (ingress from subscriber).
<ipv6-acl-out>	IPv6 downstream ACL (egress to subscriber).

4.5.3. Subscriber Filters Operational Commands

Subscriber Filters Show Commands

The show commands provide detailed information about the subscriber filter.

Subscriber ACL (Filter) Information

The show subscriber <id> acl command provides a comprehensive list of all ACL instances initiated for the subscriber, including RBFS Subscriber Filters and Ascend Data Filters (ADF). Additionally, this command provides the option to view detailed information for each filter instance by appending the corresponding filter name. Consequently, the filters are displayed with all variables, such as destination-ipv4-subscriber-prefix, replaced by their actual prefixes for clarity.

Syntax:

show subscriber <subscriber-id> **acl** <acl-name>

Example:

```
supervisor@rtbrick: op> show subscriber 13693757616973414411 acl ipv4-acl-in-ipoe-
lag-1/13693757616973414411
Rule: ipv4-acl-in-ipoe-lag-1/13693757616973414411
  ACL type: l3v4
  Ordinal: 1
  Priority: 1001
    Match:
      Direction: ingress
      Destination L4 port: 80
      IP protocol: TCP
    Action:
      HTTP-redirect: True       URL: www.rtbrick.com
```

4.6. Lawful Interception

4.6.1. Lawful Interception Overview

Lawful Interception (LI) is a legal requirement in most of the countries. It enables the legal authorities to obtain communications network data for analysis or evidence. It is a method of intercepting certain data-streams of end-users in both directions, and tunnel the intercepted traffic to a Mediation Device (MD) with information about direction of capture and reference to the intercepted connection.

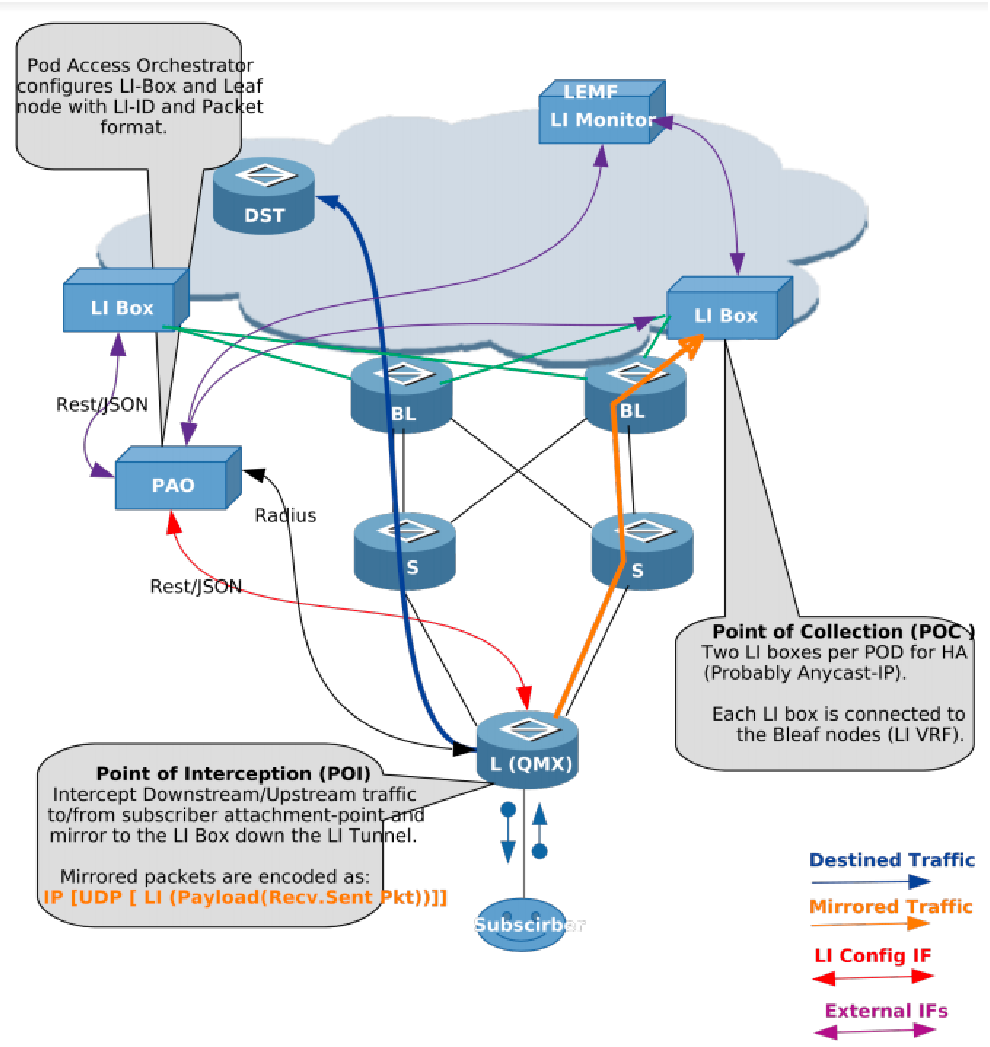
Leaf node is the Point of Interception (POI) and MD is the final Point of Collection (POC).

Supported Platforms

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

Components of Lawful Interception

The figure below shows the different components of the LI solution.


Pod Access Orchestrator configures LI-Box and Leaf node with LI-ID and Packet format.
LEMF
LI Monitor
DST
LI Box
LI Box
Rest/JSON
BL
BL
PAO
Radius
S
S
Rest/JSON
Point of Collection (POC)
Two LI boxes per POD for HA (Probably Anycast-IP).
Each LI box is connected to the Bleaf nodes (LI VRF).
L (QMX)
Point of Interception (POI)
Intercept Downstream/Upstream traffic to/from subscriber attachment-point and mirror to the LI Box down the LI Tunnel.
Mirrored packets are encoded as:
IP [UDP [LI (Payload(Recv.Sent Pkt))]]
Subscirber
Destined Traffic
Mirrored Traffic
LI Config IF
External IFs

Definitions

L(QMX)

Leaf node in the POD which is connected to subscribers.

S/BL

Spine and Border Leaf in the POD, which can be replaced with just one node.

LI Box

Lawful Intercept Box, which communicates to Law Enforcement Agency (LEMF) and relays mirrored traffic. Two LI boxes per POD are connected for redundancy.

PAO

POD Access Orchestrator, which configures the LI Box and network nodes with LI configurations.

DST

Destination node for traffic from subscribers.

Abbreviations

Abbreviation	Definition
LI	Lawful Interception
POI	Point of Interception
POC	Point of Collection
PAO	Pod Access Orchestrator
LIMS	Lawful Interception Management System
VRF	Virtual Routing Instance
LEMF	Lawful Enforcement Monitoring Facility
Leaf	Access node
PPPoE	Point to Point Protocol over Ethernet
L2TP	Layer 2 Tunnelling Protocol
MPLS	Multi Protocol Label Switching

Guidelines & Limitations

- The unidentified LI traffic is subject to the following limitations when using more than seven UDP ports.
  Currently, there is a restriction on UDP destination ports, which are limited to 7. The IP destination addresses (IP1 through IPn) can utilize any of the seven ports. The distribution of these seven ports is determined by the order in which requests are received, with priority given to those who arrive first.
- All upstream packets, regardless of whether they were dropped or not, are intercepted and mirrored to the LI collection entity.
  The following are some of the reasons that could cause dropped packets, but LI will still intercept and mirror traffic to LI collection.

  A routing failure occurred. This is unlikely as there is a default route to the spine.

  The RPF check has failed.

  The policer was dropped.

The ACL/filter was dropped.

This limitation does not apply to downstream packets.

4.6.2. LI Encapsulation

Qumran-MX (BCM) supports LI with 32 bits shim header: SHIMoUDPoIPoETH

Packet Format Encoding

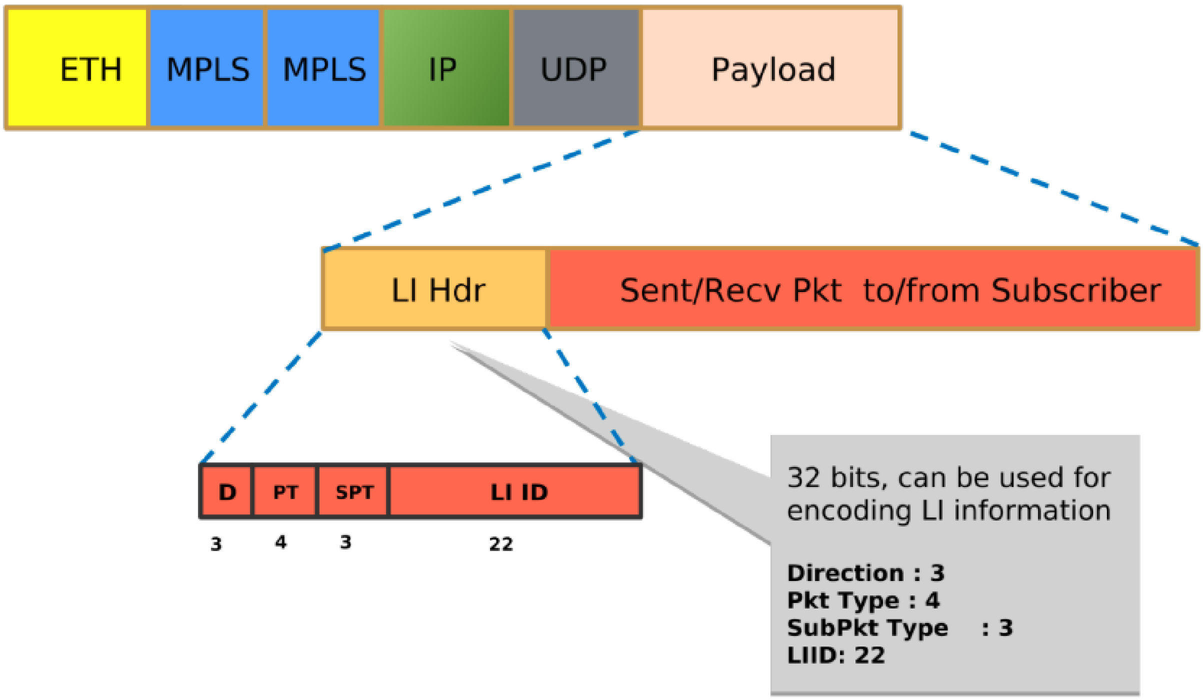


Figure 22. Packet Format Encoding

Payload Direction

Value	Payload Direction
0	Reserved for keepalive mechanism
2	Intercepted data or event was sent to target (downstream)
3	Intercepted data or event was sent from target (upstream)

Mapping Payload Format

Value	Payload Format
0-3	Reserved (unused)

Value	Payload Format
4	Unknown, Not able to decide the PT
5	IPv4 packet (not used)
6	IPv6 packet (not used)
7	Ethernet Frame (used for Lawful Interception)

Sub-payload Format (Type)

The sub-payload formats are:

1. Single VLAN tag
2. Double VLAN tag
3. Untagged

4.6.3. Enabling Lawful Interception

- RBFS hides information about lawful intercepts from all but the most privileged users. An administrator must set up access rights to enable privileged users to access lawful intercept information.
- LI can be enabled for both L2TP and PPPoE subscribers.

RADIUS Lawful Interception

All of the following attributes must be present in RADIUS access-accept or CoA request to control Lawful Interception (LI) via RADIUS. Those attributes are salt encrypted using the algorithm described in RFC 2868 for the Tunnel-Password. This encryption algorithm is defined for RADIUS access-accept messages only. To support CoA requests the request authenticator should be replaced with 16 zero bytes which is common industry standard.

RFC and draft compliance are partial except as specified.

The LI action NOOP can be used to obfuscate lawful interception requests (fake requests) to prevent that just the presence of those attributes indicates that a subscriber is intercepted. LI requests via RADIUS will show up in the same table as requests via REST or HTTP RPC API (secure.lawful.access.1.li_request).

The failed LI activations are not signalled via RADIUS to prevent that just the presence of CoA response NAK shows that LI request is not fake (action NOOP).

VSA 26-50058-140 - RtBrick-LI-Action (salt encrypted integer)

Value	Code	Description
NOOP	0	No action / Ignore LI request
ON	1	Start LI / Add LI request
OFF	2	Stop LI / Delete LI request

VSA 26-50058-141 - RtBrick-LI-Identifier (salt encrypted integer)

Device unique lawful interception identifier (LIID) within the range from 1 to 4194303.

VSA 26-50058-142 - RtBrick-LI-Direction (salt encrypted integer)

Value	Code	Description
INGRESS	1	Ingress mirroring only (from subscriber)
EGRESS	2	Egress mirroring only (to subscriber)
BOTH	3	Bidirectional mirroring (from and to subscriber)

VSA 26-50058-143 - RtBrick-LI-MED-Instance (salt encrypted string)

Routing instance through which the mediation device is reachable.

VSA 26-50058-144 - RtBrick-LI-MED-IP (salt encrypted IPv4 address)

IPv4 address of the mediation device.

VSA 26-50058-145 - RtBrick-LI-MED-Port (salt encrypted integer)

UDP port between 49152 and 65535 set in the mirrored traffic

RBFS Operational State API

The RBFS Operational State API provides endpoints for enabling and disabling LI on a per-subscriber basis:

- A HTTP POST request to /subscribers/{subscriber_id}/enableLI?id={li_id}&direction={li_direction}&med_ip={med_ip}&med_instance={med_instance}&med_port={med_port} enables LI for the specified subscriber
- A HTTP POST request to /subscribers/{subscriber_id}/disableLI?id={li_id} disable LI for the specified subscriber

The table below lists the request parameters:

Parameter Name	Description
subscriber_id	Subscriber identifier that is generated by RBFS, for example, 72339069014638701.
id	Identifier for Lawful Interception. This is unique Identifier used by mediation device to identify the intercepted subscriber. The range can be between 1 to 4194303.
direction	LI direction. Values are: INGRESS, EGRESS, BOTH.
med_instance	VRF instance through the which the mediation device is reachable.
med_ip	IPv4 address of the mediation device
med_port	UDP port(MD)(49152-65535), mirrored traffic is forwarded

> ℹ All parameters are mandatory to enable LI.

Request Examples

Enabling LI

The example below shows a curl command to enable LI:

```
curl -i -H "Content-Type: application/json" -X POST -d
http://198.51.100.76:19091/api/v1/rbfs/elements/rtbrick/services/opsd/proxy/subscr
ibers/72339069014639042/enableLI?id=66666&direction=BOTH&med_instance=libox&med_ip
=10.0.0.1&med_port=49153
```

Disabling LI

The example below shows a curl command to disable LI.

```
curl -i -H "Content-Type: application/json" -X POST -d
http://198.51.100.76:19091/api/v1/rbfs/elements/rtbrick/services/opsd/proxy/subscr
ibers/72339069014639042/disableLI?id=66666
```

4.7. L2BSA

4.7.1. L2BSA Overview

Layer 2 Bitstream Access (L2BSA) refers to a scenario in which a service provider makes his access infrastructure available to other service providers. These are retail service providers who provide their Internet services. In Germany, this service is mandated by the Federal Network Agency (German: Bundesnetzagentur or BNetzA) which is the regulatory office for electricity, gas, telecommunications, post, and railway markets. It is a federal agency of the Federal Ministry for Economic Affairs and Energy and is headquartered in Bonn, Germany.

The definition of the L2BSA service was defined by the so-called NGA Forum, an advisory board founded in May 2010 by the Bundesnetzagentur for promoting dialogue between the Bundesnetzagentur, network operators, manufacturers, states and local authorities on NGA rollout.

https://www.bundesnetzagentur.de/EN/Areas/Telecommunications/Companies/MarketRegulation/NGAForum/NGAForum_node.html

The L2BSA specification defines two interfaces, the so-called U interface (User Interface) at the customer location and the A10-NSP interface (A10 Network Service Provider) between the service provider networks. Those interface types were originally introduced in the Broadband Forum TR-101 (Migration to Ethernet-based Broadband Aggregation).

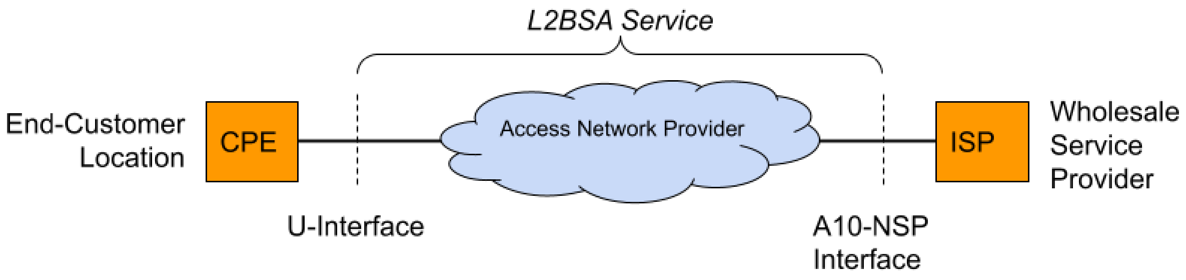


Figure 23. L2BSA Interfaces

The U interface is defined as a transparent Layer 2 interface which can be used with or without VLAN tags by the wholesale service providers. This means that some CPE will send their traffic untagged while another CPE is configured for tagged traffic which needs to be transparently forwarded between the U interface and the A10-NSP interface.

The A10-NSP interface is defined as a link aggregation bundle interface with one or more interfaces and LACP enabled or disabled. All traffic on this interface is at least single-tagged with the so-called S-VLAN tag which identifies the U interface. This limits the amount of L2BSA services to 4094 per A10-NSP interface caused by the usable VLAN range. Therefore some providers need multiple A10-NSP interfaces if they need to address more than the 4094 services.

The term A10 relates to the end-to-end ADSL network reference model depicted in the figure below. The Core Network reference model is a subset of the end-to-end architecture; it is composed of two functional blocks and three reference points. The Access Node and the Regional Broadband Network are the two functional blocks. U, V and A10 are the three reference points.

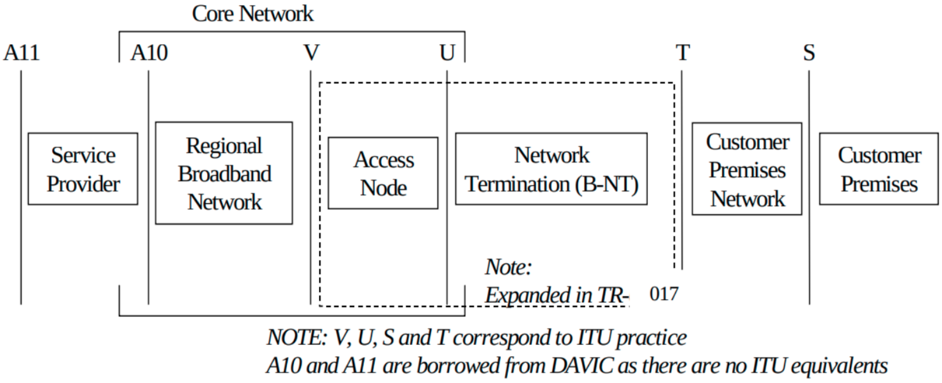

Core Network
A11
A10
V
U
T
S
Service Provider
Regional Broadband Network
Access Node
Network Termination (B-NT)
Customer Premises Network
Customer Premises
Note: Expanded in TR- 017
NOTE: V, U, S and T correspond to ITU practice
A10 and A11 are borrowed from DAVIC as there are no ITU equivalents

Figure 24. TR-025

Source: https://www.broadband-forum.org/download/TR-025.pdf

The mapping between the U interface and A10-NSP/S-VLAN is managed by the L2BSA service provider and typically changes frequently triggered by re-provisioning actions. Therefore all PPPoE discovery, as well as DHCPv4/v6 packets, must be enriched with additional line identification headers (Agent-Remote-Id, Agent-Circuit-Id, Actual-Data-Rate, …) by the L2BSA service provider in the upstream direction (from U to A10-NSP interface) to allow the wholesale provider to identify the actual U interface for traffic received on the A10-NSP interface. This functionality is referred to as the intermediate agent functionality.

The U interface is terminated on the access node which might be an OLT for fiber-based access or MSAN for xDSL, therefore the U interface can also be called the customer access line or port. The access node will add additional VLAN for each access line to uniquely identify them between the access node and leaf switch. This VLAN is called the ANP tag (Access-Node-Port).

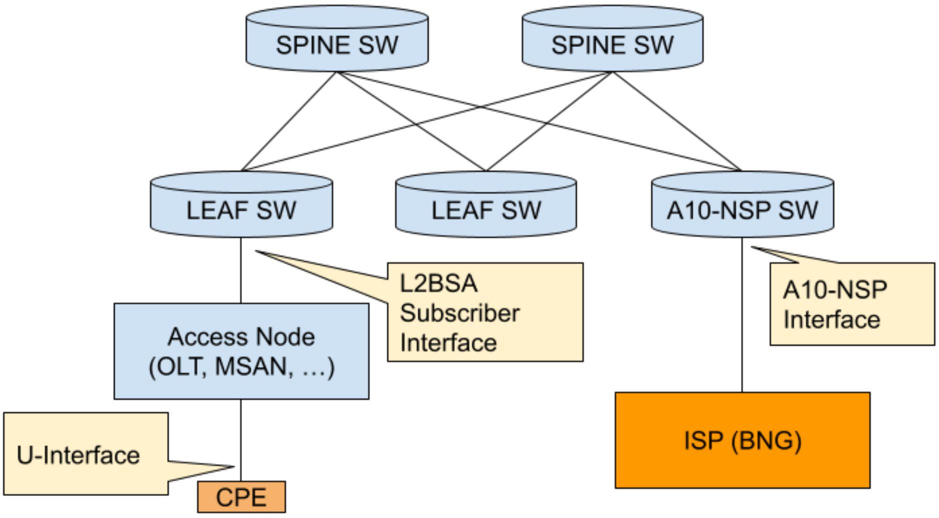


Figure 25. L2BSA Fabric

For L2BSA services all traffic received with a given ANP tag will be transparently forwarded to the corresponding A10-NSP interface via Layer 2 cross-connect (L2X) services. We introduce a new subscriber type for L2BSA in our RBFS access infrastructure to manage those services on the leaf switches and to allow further advanced access services like QoS or subscriber accounting.

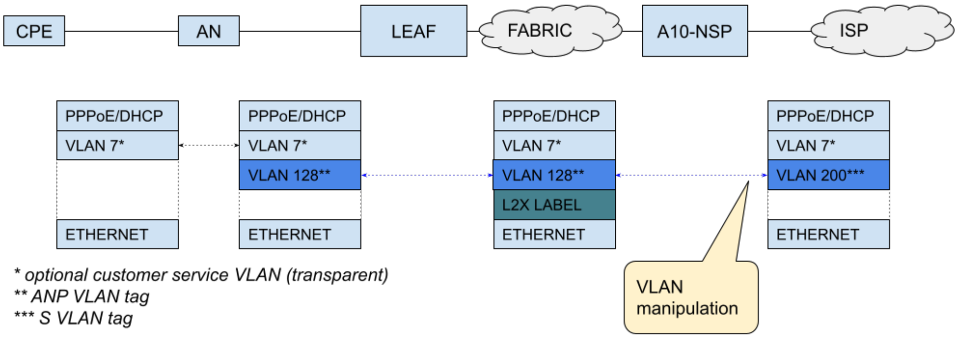


Figure 26. L2BSA Traffic

L2BSA Services and Subscribers

RBFS distinguishes between L2BSA services and subscribers. L2BSA services are created using the Operational State API (/l2bsa) stored as ephemeral objects in the configuration daemon. Those objects are excluded from the actual configuration and not restored after reboot. The L2BSA service represents a defined interface to add, update and delete L2BSA services. Those services are managed by external orchestrators or service managers and are not changed or deleted by RBFS.

Adding such an L2BSA service triggers the creation of a corresponding L2BSA subscriber which represents the internal state. Each L2BSA subscriber is a full-access subscriber and therefore most subscriber features will work for L2BSA as well. This includes features like RADIUS authentication, accounting, CoA or dynamic QoS. Each L2BSA subscriber persists as long as the service is still present and also if the subscriber has already been terminated. This allows external applications to collect all states and counters before the subscriber is finally deleted triggered by L2BSA service delete.

A10-NSP L2X Services

For each L2BSA service, there must also be an A10-NSP L2X endpoint created using the Operational State API (/a10nsp/l2x).

The A10-NSP endpoint is the place where all VLAN manipulation is done meaning where ANP and S-VLAN are swapped.

Supported Platforms

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

4.7.2. L2BSA Configuration

This document assumes a working base configuration explaining the L2BSA specific additions only.

AAA Profile Configuration

Each L2BSA service refers to a mandatory AAA configuration profile as explained in detail in the Subscriber Management Configuration Guide. The following example shows the minimum configuration with authentication and accounting disabled.

```
supervisor@leaf1: cfg> show config access aaa-profile aaa-l2bsa-default
{
    "rtbrick-config:aaa-profile": [
      {
        "profile-name": "aaa-l2bsa-default",
        "authentication": {
          "order": "NONE"
        },
        "accounting": {
          "order": "NONE"
        }
      }
    ]
  }
```

Intermediate Agent Configuration

DEPRECATED!

The functionality of the intermediate agent is designed to be implemented within the access node, which could be an OLT for fiber-based access or MSAN for xDSL. RBFS is technically prepared to fulfill the role of an intermediate agent in exceptional circumstances where the access node does not offer this function.

Please note that the intermediate agent function is deprecatd and will be removed in future releases!

The RBFS intermediate agent is capable of appending headers in the upstream direction but cannot strip them again in the downstream. Consequently, it is compatible only with PPPoE and DHCPv4. For DHCPv4, it should be noted that headers added upstream are not removed downstream if echoed back by the DHCP server, which is the usual behavior.

To enable the intermediate agent function, it must be enabled for each physical interface used by L2BSA services.

```
supervisor@leaf1: cfg> set access l2bsa intermediate-agent interface ifp-0/1/23
  <cr>
  pppoe-enable          Enable/disable the L2BSA intermediate agent for PPPoE
Discovery
  dhcpv4-enable         Enable/disable the L2BSA intermediate agent for DHCPv4
  vlan-mode             L2BSA intermediate agent VLAN mode
  inner-vlan-min        Inner VLAN min
  inner-vlan-max        Inner VLAN max
```

Attribute	Description
pppoe-enable	This option allows enabling/disabling the intermediate agent function for PPPoE discovery packets. **Default:** false (disabled)
dhcpv4-enable	This option allows enabling/disabling the intermediate agent function for DHCPv4 packets. **Default:** false (disabled)
vlan-mode	This option enables the intermediate agent function for single, double, or single and double-tagged packets. **Default:** SINGLE_DOUBLE **Values:** SINGLE, DOUBLE, SINGLE_DOUBLE
inner-vlan-min/max	This option is applicable for double-tagged packets only and allows limiting the intermediate agent function to a given inner VLAN range. **Default:** all

The following common example enables the intermediate agent for all single-tagged PPPoE/DHCPv4 packets and double-tagged PPPoE/DHCPv4 packets with inner VLAN between 1 and 3000.

```
supervisor@leaf1: cfg> show config access l2bsa intermediate-agent
{
    "rtbrick-config:intermediate-agent": {
      "interface": [
        {
          "interface-name": "ifp-0/1/23",
          "inner-vlan-min": 1,
          "inner-vlan-max": 3000,
          "pppoe-enable": "true",
          "dhcpv4-enable": "true"
        }
      ]
    }
  }
```

4.7.3. L2BSA Operational Commands

The following commands allow to list or clear all L2BSA services from CLI.

```
supervisor@leaf1: op> show l2bsa service
Interface        ANP VLAN     Timestamp
ifp-0/0/1     1000         Mon Aug 30 09:21:14 GMT +0000 2021

supervisor@leaf1: op> clear l2bsa service all
Success
```

L2BSA subscribers can be managed like any other subscriber using the subscriber CLI commands.

```
supervisor@leaf1: op> show subscriber
Subscriber-Id          Interface        VLAN      Type   State
281479271677954        ifp-0/1/23     1000:0    L2BSA  ESTABLISHED

supervisor@leaf1: op> show subscriber filter type L2BSA
Subscriber-Id          Interface        VLAN      Type   State
281479271677954        ifp-0/1/23     1000:0    L2BSA  ESTABLISHED

supervisor@leaf1: op> show subscriber 281479271677954 detail
Subscriber-Id: 281479271677954
    Type: L2BSA
    State: ESTABLISHED
    Created: Mon Aug 30 09:23:34 GMT +0000 2021
    Interface: ifp-0/1/23
    Outer VLAN: 1000
    IFL: l2bsa-0/0/1/281479271677954
    Username: ifp-0/1/23:1000@l2bsa
    Agent-Remote-Id: DEU.L2BSA.API01
    Agent-Circuit-Id: 0.0.0.0/0.0.0.0 eth 01
    AAA-Profile: aaa-l2bsa-default
    Session-Timeout: 0 (disabled)
    Idle-Timeout: 0 (disabled)
    L2X:
        Instance: default
        Nexthop: 198.51.100.101 (ipv4/labeled-unicast)
        Ingress Label (TX): 1337
        Egress Label (RX): 7331
    Accounting:
        Session-Id: 281479271677954:1630315414
        Start-Time: 2021-08-30T09:23:34.847166+0000
        Interims Interval: 0 seconds
```

The CLI also provides some global intermediate agent function packet statistics about traffic received in the control plane which belong to L2BSA subscribers.

```
supervisor@leaf1: op> show l2bsa intermediate-agent statistics
PPPoE Discovery Sent        : 0
PPPoE Discovery Updated     : 0
PPPoE Discovery Dropped     : 0
```

The sent statistics count all intermediate agent traffic forwarded back to L2X while updated only counts those which are also updated or enriched by access line information. Dropped are those packets received but dropped by the L2BSA subscriber state which might be the case if the L2BSA subscriber is still present but in a terminating state.

4.7.4. L2BSA Operational State API

The API is split into the L2BSA and A10-NSP L2X service API. Details about those API endpoints, schema and attributes can be found in the official RBFS Operational State OpenAPI documentation.

L2BSA Services API

Each L2BSA service is uniquely identified by physical interface name (IFP) and ANP VLAN identifier. All other attributes of L2BSA services can be changed dynamically by replacing the existing service object by doing a full update meaning that updates must include all attributes and RBFS will automatically recognize the changes to the last version to apply those incrementally.

Endpoint	Description
GET /l2bsa	This endpoint returns a list of all L2BSA services.
PUT /l2bsa	This endpoint replaces all L2BSA services on the system by adding new ones, updating existing and deleting those not included in the request but present on the system. Therefore this endpoint can be used to delete all services if an an empty list is provided in the request body.
GET /l2bsa/{ifp_name}	This endpoint returns a list of all L2BSA services on the given physical interface (IFP).
PUT /l2bsa/{ifp_name}	This endpoint works similarly to PUT /l2bsa but affects only services matching the given physical interface (IFP). Therefore this endpoint can be used to delete all services of a given physical interface if an empty list is provided in the request body.
GET /l2bsa/{ifp_name}/{anp}	This endpoint returns a single L2BSA service or 404 if no matching service is found.

Endpoint	Description
PUT /l2bsa/{ifp_name}/{anp}	This endpoint adds or updates a single L2BSA service.
DELETE /l2bsa/{ifp_name}/{anp}	This endpoint deletes a single L2BSA service.

The following example shows an L2BSA service request body with only the mandatory attributes set.

```
{
    "ifp_name": "ifp-0/0/1",
    "anp_vlan": 128,
    "aaa_profile_name": "aaa-l2bsa-default",
    "l2x": {
      "ingress_nexthop": "2001:db8:0:30::",
      "ingress_service_label": 1001,
      "egress_service_label": 1002
    }
}
```

The L2X ingress next-hop is typically the fabric loopback IPv4 or IPv6 address of the corresponding switch where the corresponding A10-NSP L2X service is placed. The ingress label is the upstream service label added for traffic sent to the corresponding next-hop (sent label). The egress label is the downstream service label where traffic sent from the next-hop is expected to be received (receive label). This label must be unique per switch.

Only labels greater than 100.000 should be used for the L2X service labels because labels between 16 - 100.000 are reserved by other applications. L2X service labels must not conflict with other L2X service labels and the same label can be used for downstream and upstream.

L2BSA supports also dynamic QoS which can be controlled by RADIUS if enabled or directly via API as shown in the following example.

```
{
    "ifp_name": "ifp-0/0/1",
    "anp_vlan": 128,
    "aaa_profile_name": "aaa-l2bsa-default",
    "l2x": {
      "ingress_nexthop": "2001:db8:0:30::",
      "ingress_service_label": 1001,
      "egress_service_label": 1002
    },
    "qos": {
      "qos_profile_name": "l2bsa-qos-default",
      "parent_scheduler": "pon1",
      "shaper":
"name=shaper_session,high=14000,low=2000;name=shaper_voice,high=2000",
      "policer": "level=4,cir=1m,cbs=256;level=3,cir=2m,cbs=512",
      "queue": "name=BestEffort,size=65000;name=Voice,size=20000"
    }
}
```

The QoS configuration profile is used to instantiate the QoS resources for the subscriber including schedulers, queues and shapers. This profile can be set using the qos_profile_name attribute. All dynamic QoS settings like MFC, queue sizes, shaper and policer rates will be reset if this attribute is present also if not changed. Additional shaper or policer settings included will be applied to the new QoS configuration after reset.

The assigned shaper instances can be updated using the shaper attribute which will apply to the QoS instance of the corresponding subscriber-only, but not to the other subscribers using the same QoS profile. It is only possible to update existing shapers dynamically but it is not possible to create a new shaper.

```
"shaper": "<shaper-name>,<high-kbps>,<low-kbps>;<shaper-name>,…"
```

This attribute can be also used as a key-value list which is automatically recognized by RBFS.

```
"shaper": "name=<shaper-name>,high=<high-kbps>,low=<low-kbps>;…"
```

The policer attribute is also a string that contains a list of multiple policer-level settings separated by a semicolon. Each set contains a level, cir, cbs, pir, pbs, max-cir and max-pir separated by a comma.

```
"policer": "<level>,<cir>,<cbs>,<pir>,<pbs>,<max-cir>,<max-pir>;<level>…"

Example:
"policer": "1,2000,200;2,8000,800;3,0,800;4,0,800"

level: 1 (highest priority) to 4 (lowest priority)
cir: ingress policer committed information rate (kbps)
cbs: ingress policer committed burst size (kbps)
pir: ingress policer peak information rate (kbps)
pbs: ingress policer peak burst size (kbps)
max-cir: max ingress policer committed information rate (kbps)
max-pir: max ingress policer peak information rate (kbps)
```

If PIR and PBS are not defined, the values from CIR and CBS are used as PIR and PBS as well. The max CIR and max PIR attributes are optional default set to unlimited.

This attribute can be also used as a key-value list which is automatically recognized by RBFS.

```
"policer": "level=<level>,cir=<cir>,cbs=<cbs>,pir=<pir>,pbs=<pbs>,max-cir=<max-
cir>,max-pir=<max-pir>;…"

Example:
"policer": "level=4,cir=1m,cbs=256;cir=2m,cbs=512,level=3"
```

The queue attribute is a string that contains a list of multiple queue settings separated by a semicolon. Each queue setting contains a queue name and queue size in bytes separated by a comma.

```
"queue": "<queue-name>,<size-bytes>;<queue-name>,<size-bytes>;…"
```

This attribute can be also used as a key-value list which is automatically recognized by RBFS.

```
"queue": "name=<queue-name>,size=<size-bytes>;name=…"
```

The subscriber management infrastructure does not check if a referenced QoS profile, shaper, queue or policer exists or not. This is handled by forwarding infrastructure which continues processing the subscriber QoS settings as soon as the missing resource was added.

The parent scheduler element of the scheduler map assigned to the subscriber can be selected with the parent_scheduler attribute. If not present, the scheduler map will be directly bound to the local IFP where the session is established. The parent scheduler can be only set at the beginning and must not be changed.

Providing a QoS parent scheduler that is not present on the corresponding IFP will lead to the black howling of all egress data traffic.

The optional access_line_info attribute contains the access line identification and characteristics attributes used by the intermediate agent function. A detailed list of all supported line attributes and values can be found in the official RBFS Operational State OpenAPI documentation.

> For access line attributes there is a difference between attributes not present or set to zero. Those not present will be also not added by the intermediate agent function whereas those set to zero will be added by the intermediate agent function with value also set to zero.

Example:

```
/usr/bin/curl --location --request PUT
'http://198.51.100.35:19091/api/v1/rbfs/elements/leaf1/services/opsd/proxy/l2bsa/i
fp-0%2f0%2f1/128' --header 'Content-Type: application/json' --data-raw '{
    "ifp_name": "ifp-0/0/1",
    "anp_vlan": 128,
    "aaa_profile_name": "aaa-l2bsa-default",
    "l2x": {
        "ingress_nexthop": "2001:db8:0:30::",
        "ingress_service_label": 1001,
        "egress_service_label": 1002
    },
    "qos": {
        "qos_profile_name": "l2bsa-qos-default",
        "parent_scheduler": "pon1",
        "shaper":
"name=shaper_session,high=14000,low=2000;name=shaper_voice,high=2000",
        "policer": "level=4,cir=1m,cbs=128;level=2,cir=1m,cbs=128",
        "queue": "name=BestEffort,size=65000;name=Voice,size=20000"
    },
    "access_line_info": {
        "agent_circuit_id": "0.0.0.0/0.0.0.0 eth 0/0",
        "agent_remote_id": "DEU.RTBRICK.L2BSA01",
        "upstream": {
            "actual_rate": 2048
        },
        "downstream": {
            "actual_rate": 16235
        },
        "pon_type": "GPON"
    }
}'
```

L2BSA Subscribers API

The existing subscriber API was extended to do some common actions using the L2BSA physical interface name (IFP) and ANP VLAN identifier instead of the subscriber-id.

Endpoint	Description
GET /subscribers/l2bsa/{ifp_name}/{anp}	This endpoint is an alias for GET /subscribers/{subscriber_id} returning detailed subscriber information or 404 if no matching subscriber is found.
DELETE /subscribers/l2bsa/{ifp_name}/{anp}	This endpoint is an alias for DELETE /subscribers/{subscriber_id} which terminates a matching L2BSA subscriber.
GET /subscribers/l2bsa/{ifp_name}/{anp}/adjusted-accounting	This endpoint is an alias for GET /subscribers/{subscriber_id}/adjusted-accounting returning the adjusted accounting information or 404 if no matching subscriber is found.

A10-NSP L2X Services API

Endpoint	Description
GET /a10nsp/l2x	This endpoint returns a list of all A10-NSP L2X services.
PUT /a10nsp/l2x	This endpoint replaces all A10-NSP L2X services on the system by adding new ones, updating existing and deleting those not included in the request but present on the system. Therefore this endpoint can be used to delete all services if an an empty list is provided in the request body.
GET /a10nsp/l2x/{lag_interface_name}	This endpoint returns a list of all A10-NSP L2X services on the given the LAG interface.
PUT /a10nsp/l2x/{lag_interface_name}	This endpoint works similarly to PUT /a10nsp/l2x but affects only services matching the given LAG interface. Therefore this endpoint can be used to delete all services of a given the LAG interface if an empty list is provided in the request body.
GET /a10nsp/l2x/{lag_interface_name}/{s_anp}	This endpoint returns a single A10-NSP L2X service or 404 if no matching service is found.

Endpoint	Description
PUT /a10nsp/l2x/{lag_interface_name}/{s_anp}	This endpoint adds or updates a single A10-NSP L2X service.
DELETE /a10nsp/l2x/{lag_interface_name}/{s_anp}	This endpoint deletes a single A10-NSP L2X service.

The following example shows an A10-NSP L2X service request body with only the mandatory attributes set.

```
{
    "lag_interface_name": "lag-1",
    "s_vlan": 100,
    "anp_vlan": 128,
    "l2x": {
        "ingress_nexthop": "2001:db8:0:40::",
        "ingress_service_label": 1002,
        "egress_service_label": 1001
    }
}
```

The L2X ingress next-hop is typically the fabric loopback IPv4 or IPv6 address of the corresponding switch where the corresponding L2BSA service is placed. The ingress label is the upstream service label added for traffic sent to the corresponding next-hop (send label). The egress label is the downstream service label where traffic sent from the next-hop is expected to be received (receive label). This label must be unique per switch.

> Only labels greater than 100000 should be used for the L2BSA service labels because labels between 16 - 100000 are reserved by other services. L2BSA service labels must not conflict with other L2X service labels and the same label can be used for downstream and upstream.

Example:

```
/usr/bin/curl --location --request PUT
'http://198.51.100.35:19091/api/v1/rbfs/elements/a10nsp-
sw1/services/opsd/proxy/a10nsp/l2x/lag-1/100' \
--header 'Content-Type: application/json' \
--data-raw '{
    "lag_interface_name": "lag-1",
    "s_vlan": 100,
    "anp_vlan": 128,
    "l2x": {
        "ingress_nexthop": "2001:db8:0:40::",
        "ingress_service_label": 1002,
        "egress_service_label": 1001
    }
}'
```

- L2BSA supports both PPPoE and IPoE subscribers.
- L2BSA supports DHCPv4 and DHCPv6 protocols.

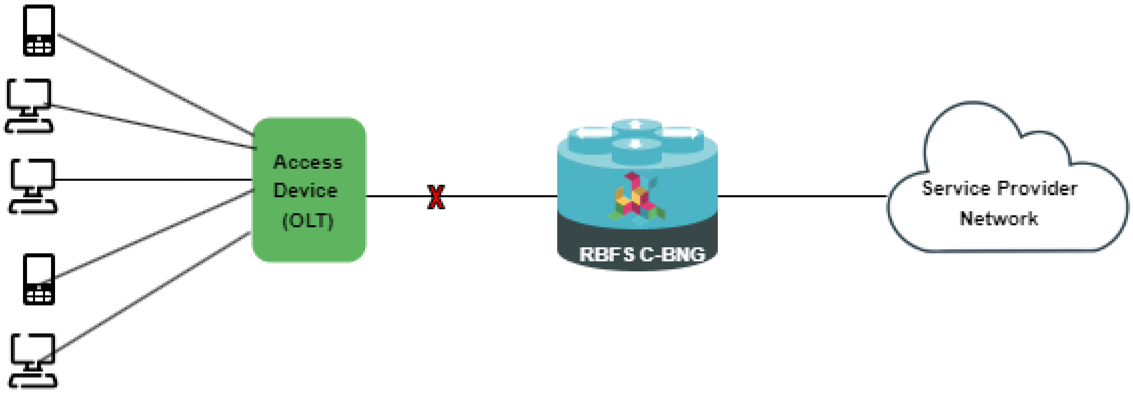
4.8. Redundancy

4.8.1. RBFS Redundancy for Subscriber Groups

Overview

Node outages and link failures that may occur on an access network can bring down the subscriber services. These network outages affect critical workloads and continuity of business. So, it is essential to set up a network that is resilient and responds quickly to the events and protect the network from outages.

The following diagram represents a simple access network without redundancy.



RBFS Redundancy protects subscriber services from node or 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 goes down due to node or link failure, another node can automatically take over the services.

Currently, RBFS Redundancy supports only IPoE.

Understanding RBFS Redundancy

RBFS Redundancy protects subscriber groups using an active-standby node cluster model. In the active-standby node cluster, the active node (for a subscriber group) performs subscriber services. The standby device mirrors concurrent subscriber state data from the active peer (for that redundancy session). Both the nodes, paired for redundancy, keep sending 'keepalive' messages to each other to check the health status. RBFS Redundancy is centered around subscriber groups, known as redundancy sessions.

> The document uses the term C-BNG throughout the document. C-BNG stands for consolidated BNG. Unlike the spine-leaf topology, where the functionalities are distributed to spine platforms and leaf platforms separately based on their roles, C-BNG platform includes all functionalities together in a single platform.

> RBFS Redundancy is not supported on spine-leaf network topology as the RBFS spine-leaf topological architecture itself innately provides a redundant and resilient network.

Inter-BNG Connectivity

C-BNG 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 C-BNG pairs establishes connectivity and the link is used to send 'keepalive' messages and data mirroring for subscriber state synchronization.

If the link goes down, the Keepalive messages cannot be exchanged between the C-BNGs and the C-BNGs move to the standalone state only after the hold-timer expires and the previously synchronized subscriber session data becomes invalid.

If the C-BNG pairs are connected through the core network, then a core network failure can impact redundancy. However, if the inter-BNG connection is direct or through a different path that doesn’t rely on the core, redundancy is not impacted by core network failure.

In a scenario, when a new lag is configured for redundancy with the same redundancy session ID, it works for the same session. If any of the lag goes down, a switchover happens.

Redundancy Session

Redundancy session is a binding mechanism that is used to pair C-BNGs for redundancy. RBFS Redundancy allows grouping of subscribers, under a redundancy session and each redundancy session is represented by a redundancy session ID. Simply, an RBFS Redundancy session represents a redundancy group of subscribers. A redundancy session enables linking the LAG with that particular redundancy session (subscriber group). When you define a value for the redundancy session ID, this ID should be unique and the same for both redundancy pairs. When two nodes get the same session ID, they recognize each other as the peer nodes for a particular redundancy session (subscriber group). The TCP session establishment between the nodes occurs after the pairing with the redundancy session ID. Once the TCP session is established, the nodes use this channel for subscriber data mirroring and synchronization and for health status monitoring.

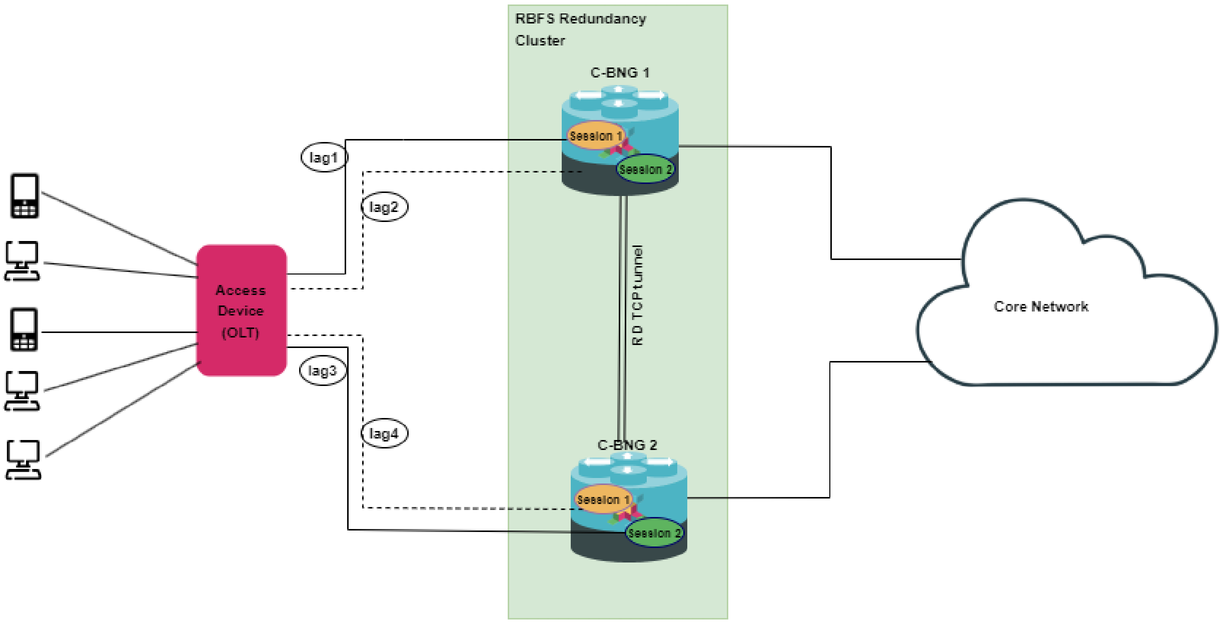
A C-BNG chassis can contain multiple redundancy sessions. Multiple C-BNG nodes can be active nodes for one or more subscriber groups (redundancy sessions) that serve the subscribers and they can, at the same time, be standby nodes for other subscriber groups.

One node, which is paired for redundancy, can perform subscriber services for more than one redundancy session (subscriber group). The peer node, which is identical to the first node, contains the same subscriber group (redundancy session) as a standby. And in the event of that first node goes down due to any outage, the standby node can take over subscriber service for this redundancy session.

RBFS Redundancy allows running a redundancy session actively on one node and back up the same redundancy session (subscriber group) on a different (standby) C-BNG node. In RBFS redundancy, in fact, there is no active node or standby node. It is active subscriber group and standby subscriber group. A C-BNG node can be active for a subscriber group and at the same time it can be a standby for a different subscriber group. You can park a maximum number of 64 redundancy sessions (either active or standby) on a C-BNG node.

RBFS Redundancy Architecture

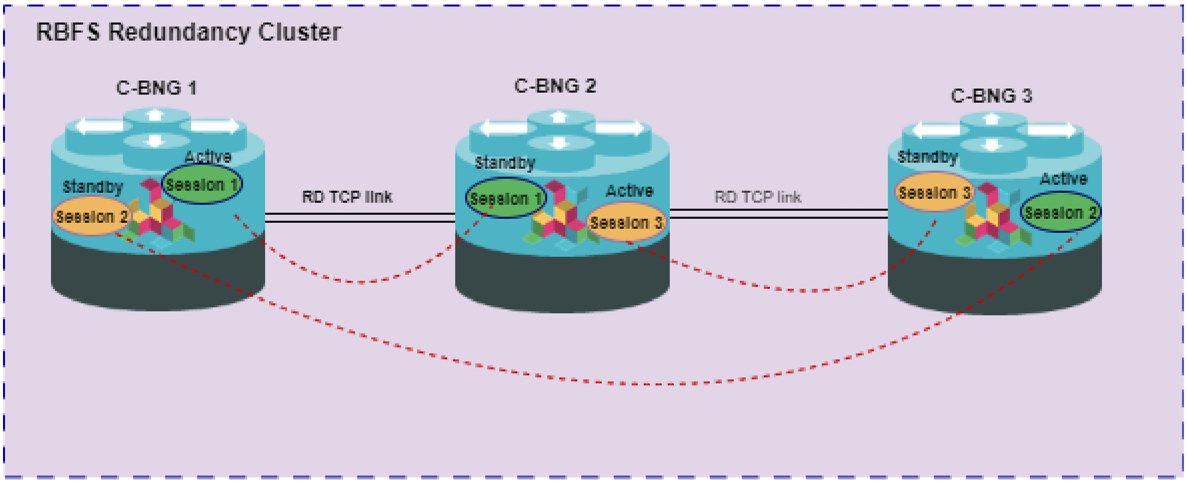
The following architectural diagram provides a high-level view of RBFS in redundancy mode. It shows two RBFS nodes, paired for redundancy, deployed in a active-standby node cluster, with their interfaces are connected with an RD TCP connection. These peer nodes use the RD TCP connection for sending 'keepalive' messages and data mirroring for subscriber state synchronization.



Both of the nodes are connected to an access device (OLT device in this scenario) on one end from where it receives subscriber traffic and sends traffic. The nodes are also connected to the core network on the other end.

The node 'C-BNG 1' is in active state and performs subscriber services for the 'Session 1' (redundancy subscriber group). 'Session 1' is also mirrored in the C-BNG 2 in standby mode. The standby device mirrors concurrent subscriber state data for 'Session 1' from the active peer. If an active node goes down due to any reason, the peer node detects the outage and uses mirrored 'Session 1' to perform subscriber services.

One C-BNG node acts as active node for one or more sessions (subscriber redundancy groups) and as a standby C-BNG for other subscriber redundancy groups at the same time. The following diagram illustrates the scenario.



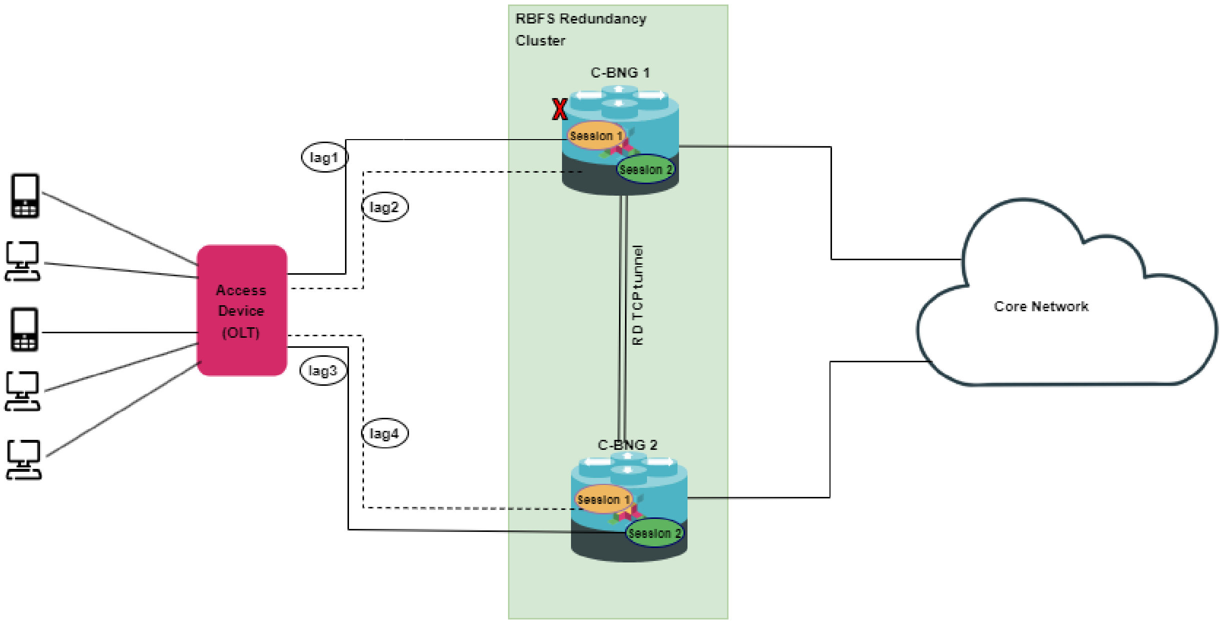
'Session 1' is active in C-BNG 1 and is in standby mode in C-BNG 2.
'Session 2' is active in C-BNG 3 and is in standby mode in C-BNG 1.
'Session 3' is active in C-BNG 2 and is standby mode in C-BNG 3.

RBFS Redundancy can mitigate the following types of failures:

- Link failure Between Active RBFS Node and Access Node (OLT, DSLAM or MSAN)
- Node Outage

Redundancy for Node Outage

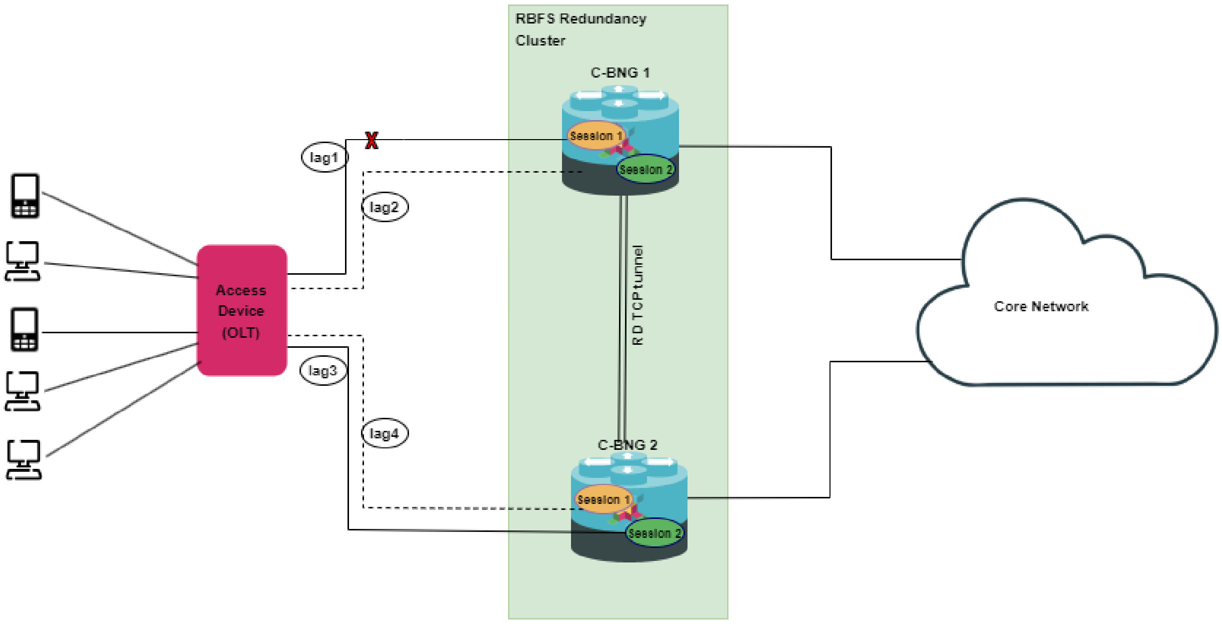
A node outage, which can bring down the subscriber services, can occur due to many reasons on a network. RBFS Redundancy helps to minimize the impact and reduce interruptions and downtime by providing a resilient system. In the event of a node outage, RBFS Redundancy triggers switchover in which the standby node takes over from the active node with very minimal impact on the subscriber services.



The diagram shows cbng1 as an active node serving subscribers and cbng2 stays as a standby. When an active node goes down, the standby node detects the same and takes over from the active RBFS node. In a node outage scenario, the node becomes unresponsive and cannot maintain communication with its peer node the RD TCP link.

Redundancy for Link Failure

In RBFS Redundancy, Link Aggregation combines multiple physical links into a single logical link. If a member link, which is part of a LAG, goes down, the LAG fails. The diagram shows the 'cbng1' and 'cbng2' deployed in redundancy mode and is connected to the access device with LAGs. When the LAG between 'cbng1' and the access node goes down, the 'cbng1', which is running 'Session 1', becomes inactive for the subscriber group. So the 'cbng2', which is the standby for 'Session 1', detects the failure of 'cbng1' and starts performing subscriber services for 'Session 1' by providing a quick recovery from the disruption.



In this link or LAG failure scenario, cbng1 is in a healthy state, only the LAG interface went down. It cannot perform subscriber services for the particular redundancy session. However, it can keep communication with its peer node as the RD TCP channel.

Subscriber Data Synchronization

RBFS Redundancy subscriber data from the active node is always synced to the standby node. So that if the active node goes down, the standby node takes over and restores traffic forwarding which was previously performed by its peer node. It ensures that traffic can keep flowing even in the event of an outage.

Node States in Redundancy

In RBFS redundancy, C-BNG nodes have different states for various redundancy sessions. Typically, RBFS redundancy nodes encounter the following states for redundancy sessions:

Active: All subscribers are served by active node in the RBFS active-standby node cluster. One node which is active for a redundancy session can be a standby node for a different session. Nodes, that are paired for redundancy, send 'keepalive' messages to each other and also synchronize all subscriber state data with the peer node. The priority values that you specify for the redundancy nodes determine the roles of active and standby. The node that receives the higher priority value for the session ID assumes the role of active for that subscriber group. To set one device as 'active', you must specify higher priority value for the redundancy session for that node.

Standby: Standby node is identical with the active node and synchronizes subscriber data concurrently from peer node. It keeps communication with the peer node to monitor node health status using 'keepalive' messages. The node that gets the lower priority value for the redundancy session ID assumes the role of standby. Standby node for a subscriber group does not perform any subscriber services for that group unless or until the active node encounters an outage.

Down: When a node becomes inactive due to an outage, it is considered as 'down'. In the event of a node outage, it is completely down and cannot perform subscriber services and any communication with its peer node. But in the case of a LAG (between the node and access node) failure, the node cannot perform subscriber services, but it can communicate with the peer node through the RD TCP connection. So that the subscriber state synchronization occurs without any interruption.

Stand Alone: When the active node goes down, the switchover occurs and standby takes over the subscriber service. In this scenario, the serving node is in 'stand alone' state (for that redundancy session) as it has no peer node for redundancy.

Revert or Rollback

In RBFS Redundancy, after a node or link failure and the subsequent switchover, the standby takes over and continues the subscriber service for that subscriber group even after the other node (previously active) recovers from the failure. There is no automated rollback or revert to the previously active router. However, administrators can perform a manual switchover.

Monitoring Node Health Status

The RBFS nodes, which have switchover capacities, monitor each other for the health status. RBFS Redundancy uses 'keepalive' messages that check on the health of the RBFS nodes. Both of the devices send 'keepalive' messages to each other in every five seconds. One Node can detect a failure if it does not receive 'keepalive' messages for a period of 20 seconds from the other node.

Redundancy Clients

There are multiple RBFS daemons that participate in providing redundancy. They include redundancy daemon, (rd), LAG daemon (lagd), interface daemon (ifmd), subscriber daemon (subscriberd), IPoE daemon (ipoed). These daemons, which perform various roles, are known as redundancy clients.

Redundancy Daemon

Redundancy Daemon is responsible for establishing high-availability connections. It monitors the ecosystem and detects any outages that may happen on the network. It assigns the roles of active or standby to the nodes depending on the priority configured on the node. The daemon triggers a switchover to the standby node if a failure occurs. It responds to the failure events which are reported locally by daemons who are the redundancy clients. It also cleanses the data after switchhover from the node that went down.

Supported Hardware Platforms

Currently, RBFS can be deployed in redundancy mode using the RBFS C-BNG (Consolidated BNG) switches. RBFS C-BNG software provides complete BNG functionalities on a single compact hardware switch. You can use the following hardware platforms to deploy RBFS in redundancy mode.

- UfiSpace S9600-72XC: The UfiSpace S9600-72XC is a multi-function, disaggregated white box aggregation routing platform that is equipped with Broadcom’s Qumran2c chipset. It features 64x25GE and 8x100GE high-speed ports with a switching capacity of up to 2.4Tbs.
- Edgecore AGR420: AGR420 is a high performance 25GbE aggregation router that consists of fixed 64 x 10G/25G SFP28, 8 x 100GE QSFP28 and 2 x 100G QSFP-DD network interface configurations.

RBFS Redundancy Requirements

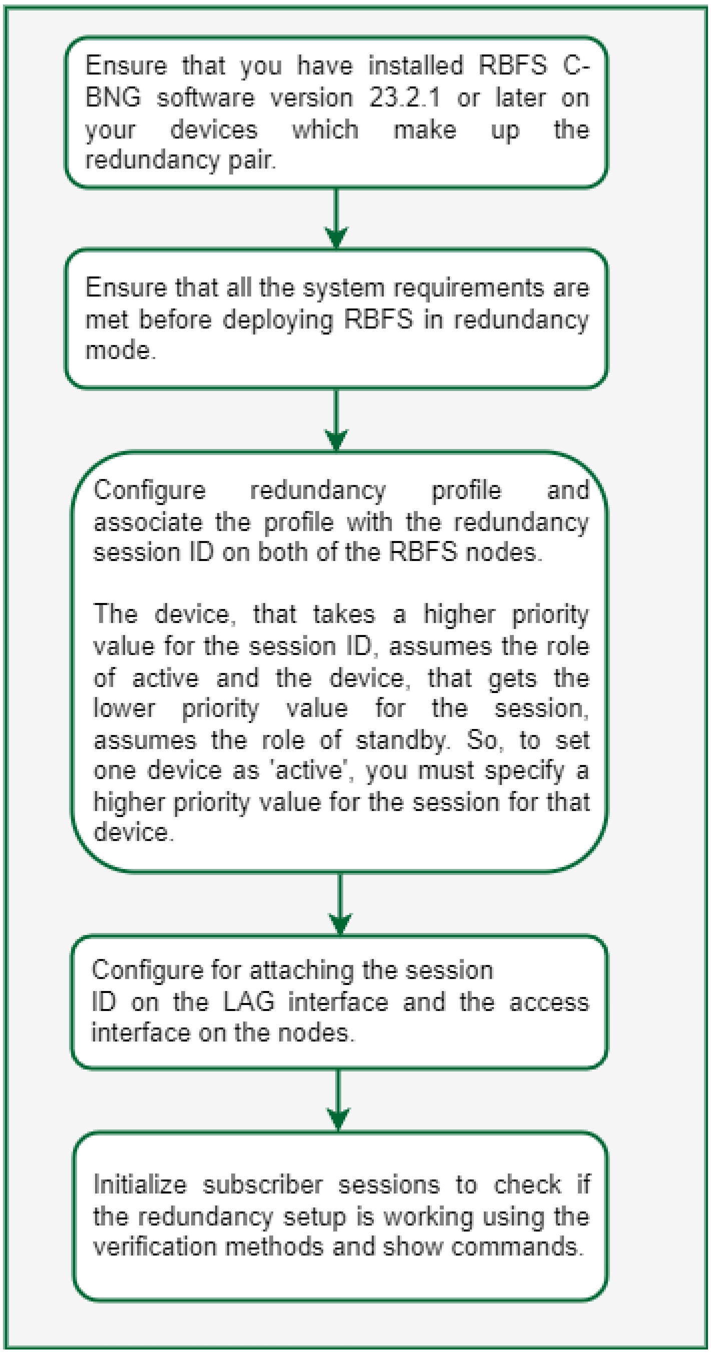
The following are the requirements for setting up RBFS Redundancy.

- Ensure that both of the platform devices, on which RBFS software runs, must be the same model.
- Ensure that the devices should run the same version of RBFS software. RBFS software 23.2.1 and later versions support deployment in redundancy mode.
- NTP must be configured on both devices to match the timestamps.

4.8.2. Deploy RBFS in Redundancy Mode

RBFS Redundancy

This section provides information on how to deploy RBFS C-BNG device pair to achieve redundancy. The following workflow diagram depicts the end-to-end deployment of RBFS in Redundancy mode:



Ensure that you have installed RBFS C-BNG software version 23.2.1 or later on your devices which make up the redundancy pair.

Ensure that all the system requirements are met before deploying RBFS in redundancy mode.

Configure redundancy profile and associate the profile with the redundancy session ID on both of the RBFS nodes.

The device, that takes a higher priority value for the session ID, assumes the role of active and the device, that gets the lower priority value for the session, assumes the role of standby. So, to set one device as 'active', you must specify a higher priority value for the session for that device.

Configure for attaching the session ID on the LAG interface and the access interface on the nodes.

Initialize subscriber sessions to check if the redundancy setup is working using the verification methods and show commands.

You must perform the following tasks to deploy the RBFS devices in redundancy mode. These configurations must be performed on both of the devices.

1. Configure Redundancy Profile
2. Configure Session for Redundancy
3. Configure Link Aggregation Group for Redundancy
4. Configure Access for Redundancy

Configuration Syntax and Commands

The following sections describe syntax and commands for various configurations.

Configuring Redundancy Profile

Redundancy profile configuration is used to provide peer identity in redundancy. While configuring the redundancy profile on the nodes, you must specify IP addresses of both the peer nodes. Based on the priority value that you specify for the session ID, the peers take the roles of active and standby for the Session.

Syntax:

set redundancy profile <name>

Attribute	Description
peer	Redundancy configuration
switchover-hold-timer	Minimum time interval between consecutive switch-overs in seconds.

Run the following commands to configure redundancy profile.

```
set redundancy profile rd_ipoe
set redundancy profile rd_ipoe peer ipv4 remote-address 198.51.100.2
set redundancy profile rd_ipoe peer ipv4 update-source 198.51.100.1
set redundancy profile rd_ipoe peer ipv4 instance default
```

Example Configuration:

```
supervisor@rtbrick>cbng1.rtbrick.net: cfg> show config redundancy profile
{
    "rtbrick-config:profile": [
      {
        "name": "rd_ipoe",
        "peer": {
          "ipv4": {
            "remote-address": "198.51.100.2",
            "update-source": "198.51.100.1",
            "instance": "default"
          }
        }
      }
    ]
  }
```

Configuring Session for Redundancy

You can configure Redundancy Session with a unique session ID. You can define the system priority value (which determines active and standby roles) and the associate Redundancy profile configuration in the Redundancy Session configuration.

Syntax:

set redundancy session <session-id>

Attribute	Description
keepalive-interval	Keepalive message transmission interval in seconds. Default is 5 seconds.
priority	Session priority
profile	Profile name

Run the following commands to configure session for redundancy.

```
set redundancy session 100
set redundancy session 100 priority 10
set redundancy session 100 profile rd_ipoe
```

Configuration Example:

```
supervisor@rtbrick>cbng2.rtbrick.net: cfg> show config redundancy session
{
    "rtbrick-config:session": [
      {
        "session-id": 100,
        "priority": 10,
        "profile": "rd_ipoe"
      }
    ]
  }
```

Configuring LAG for Redundancy

You must associate the Redundancy Session with the LAG. While configuring LAG for redundancy, you must specify the session ID to associate the LAG with the Redundancy Session. LAG can identify its Redundancy Session with this mapping.

Syntax:

set link-aggregation interface <interface-name> **options**

Attribute	Description
description	Link aggregation interface description.
member-interface	Link aggregation member interface configuration
minimum-link-count	Minimum number of active member links required for the link aggregation interface. default values is 1.
mode	Mode of the link aggregation interface, static or lacp. default mode is lacp.
rd-role	Role of the link aggregation interface, active or standby.
rd-system-priority	The value for the system priority range from 1 to 65535. The lower the value, the higher the system priority. default value is 65535.
redundancy-session-id	The value for the redundancy group session id range from 1 to 65535.
system-id	Redundancy System ID of link-aggregation interface.

Run the following commands to configure LAG for redundancy.

```
set link-aggregation interface lag-1
set link-aggregation interface lag-1 mode lacp
set link-aggregation interface lag-1 minimum-link-count 1
set link-aggregation interface lag-1 redundancy-session-id 100
set link-aggregation interface lag-1 system-id a8:b5:7e:8f:66:43
set link-aggregation interface lag-1 member-interface ifp-0/1/260
```

Example Configuration:

```
supervisor@rtbrick>cbng2.rtbrick.net:: cfg> show config link-aggregation interface
lag-1
{
    "rtbrick-config:interface": [
      {
        "interface-name": "lag-1",
        "mode": "lacp",
        "minimum-link-count": 1,
        "redundancy-session-id": 100,
        "system-id": "11:22:33:44:55:66",
        "member-interface": [
          {
            "member-interface-name": "ifp-0/0/4"
          }
        ]
      }
    ]
  }
```

Configuring Access for Redundancy

IPoE and Redundancy Session mapping is essential to associate the Redundancy Session with IPoE. While configuring access for redundancy, you must specify access type as IPoE.

Syntax:

set access interface double-tagged <name> <options>

Attribute	Description
aaa-profile-name	AAA profile name
access-profile-name	Access profile name
access-type	Access service type
gateway-ifl	IPoE gateway IFL (unnumbered source IFL)
max-subscribers-per-mac	Restrict maximum subscribers per MAC address
max-subscribers-per-vlan	Restrict maximum subscribers per VLAN
redundancy-session-id	Redundancy session id for this interface
service-profile-name	Service profile name
vlan-profile-enable	Enable VLAN profiles

Run the following commands access for redundancy.

```
set access interface double-tagged lag-1 1001 1100 1001 1100
set access interface double-tagged lag-1 1001 1100 1001 1100 access-type IPoE
set access interface double-tagged lag-1 1001 1100 1001 1100 access-profile-name
ipoe
```

```
set access interface double-tagged lag-1 1001 1100 1001 1100 aaa-profile-name
ipoe-aaa
set access interface double-tagged lag-1 1001 1100 1001 1100 gateway-ifl lo-
0/0/0/10
set access interface double-tagged lag-1 1001 1100 1001 1100 redundancy-session-id
100
```

Example Configuration:

```
supervisor@rtbrick>cbng1.rtbrick.net: cfg> show config access interface double-
tagged lag-1
{
    "rtbrick-config:double-tagged": [
      {
        "interface-name": "lag-1",
        "outer-vlan-min": 1001,
        "outer-vlan-max": 1100,
        "inner-vlan-min": 1001,
        "inner-vlan-max": 1100,
        "access-type": "IPoE",
        "access-profile-name": "ipoe",
        "aaa-profile-name": "ipoe-aaa",
        "gateway-ifl": "lo-0/0/0/10",
        "redundancy-session-id": 100
      }
    ]
  }
```

Switchover Manually

Administrators can perform a manual switchover from an active node to a standby node. Use the following command to perform switchover:

```
switch-over session <session-id> confirm
```

4.8.3. RBFS Redundancy Operational Commands

Redundancy Show Commands

With the RBFS Command Line Interface, you can view output of operational commands. The redundancy operational commands provide detailed information about the RBFS redundancy operations.

Client Statistics

Syntax:

show redundancy client <name> **statistics**

This command displays information from redundancy client (daemon) which is a participant in the redundancy sessions. The daemons ifmd, ipoed.1, lagd, and subscriberd.1 are the client daemons in redundancy sessions.

Command:

```
show redundancy client lagd statistics
```

Example:

```
supervisor@rtbrick>ufi08.q2c.u23.r4.nbg.rtbrick.net: op> show redundancy client
lagd statistics
Session id: 100, Profile: rd_ipoe
  Agent State: down
  TCP operational state: down
  Message statistics:
    Keep alive sent: 0
    Keep alive received: 0
  Last 5 state changes: [Latest first]
    down                : 2022-12-08T08:47:06.077823+0000
    demote-ready        : 2022-12-08T08:47:06.077798+0000
    demote-infra-wait   : 2022-12-08T08:47:06.077769+0000
    demote-app-wait     : 2022-12-08T08:47:06.077713+0000
    active              : 2022-12-08T06:57:53.058172+0000
  Connection statistics:
    Peer Address: 198.51.100.2
    Application down notifications: 0
    Connection down notifications: 0
    Retry count: 0
    Session down received: 8
```

Session Details

Syntax:

show redundancy session detail

The command displays RD session details.

Command:

```
show redundancy session detail
```

Example:

```
supervisor@rtbrick>ufi08.q2c.u23.r4.nbg.rtbrick.net: op> show redundancy session
detail
Redundancy session ID: 100, Update source: 198.51.100.1, Peer: 198.51.100.2.
  Instance: default, Profile name: rd_ipoe, Local priority: 20
  State: down, Previous state: active, Last state transition time: 2022-12-
08T08:47:06.071930+0000
  TCP operational state: down
  Message statistics:
    Keep alive sent: 21516
    Keep alive received: 21515
    Switch overs detected: 0
  Timers:
    Connect retry interval: 2000
    keep alive timer interval: 3000
    Holddown timer interval: 9000
```

Session ID Details

Syntax:

show redundancy session <ID> **detail**

The command displays RD session detail for a session ID. Command:

```
show redundancy session 100 detail
```

Example:

```
supervisor@rtbrick>ufi08.q2c.u23.r4.nbg.rtbrick.net: op> show redundancy session
100 detail
Redundancy session ID: 100, Update source: 198.51.100.1, Peer: 198.51.100.2
  Instance: default, Profile name: rd_ipoe, Local priority: 20
  State: down, Previous state: active, Last state transition time: 2022-12-
08T08:47:06.071930+0000
  TCP operational state: down
  Message statistics:
    Keep alive sent: 21516
    Keep alive received: 21515
    Switch overs detected: 0
  Timers:
    Connect retry interval: 2000
    keep alive timer interval: 3000
    Holddown timer interval: 9000
```

Session ID Status

Syntax:

show redundancy session <ID> **status**

The command displays status information of a session ID.

Command:

```
show redundancy session* <ID> *status
```

Example:

```
supervisor@rtbrick>ufi08.q2c.u23.r4.nbg.rtbrick.net: op> show redundancy session
100 status
State: down,  Remote State: invalid
Redundancy client replication information:
  Total redundancy clients : 5
  ifmd:
  ipoed.1:
  lagd:
  poold:
  subscriberd.1:
    Number of subscribed table: 1
```

Session History

Syntax:

show redundancy session <ID> **status**

The command displays history of a session ID for a specified count.

```
show redundancy session 100 history count 3
```

```
supervisor@C2-STD-27-2804>bm04-tst.fsn.rtbrick.net: cfg> show redundancy session
100 history count 3
Previous state       Current state        State change reason  Timestamp
connect              standby              standby              2022-12-
21T07:05:36.010847+0000
down                 connect              open                 2022-12-
21T07:05:29.738121+0000
invalid              down                 init                 2022-12-
21T07:00:44.282300+0000
```

5. Forwarding

5.1. Interfaces

5.1.1. Interfaces Overview

RtBrick Full Stack (RBFS) supports various types of interfaces, including physical and logical interfaces. On hardware platforms, RBFS physical interfaces represent the ports of a switch. This guide describes how to configure and verify RBFS interfaces. Features like routing protocols or access services will typically run on top of the interfaces.

Interface Types

Physical Interfaces

In RBFS, physical interfaces (IFP) typically represent the physical ports of a hardware switch. For example, ifp-0/0/1 represents switch port 1. On the physical interface level, you can configure various parameters associated with Layer 1 of the ISO/OSI reference model.

Logical Interfaces

For each physical interface, you can create one or multiple interface units also referred to as logical interfaces (IFL) in RBFS. A logical interface is associated with the Layer 2 operation. In addition, you can configure Layer 3 parameters like IP addresses on interface units, and assign interface units to routing instances.

Loopback Interfaces

A loopback interface is typically used to represent and identify a device itself. Loopback interfaces are preferred because they do not depend on the status of a physical port, and will always be up. Please note, although loopback interfaces are virtual interfaces, there are also represented as physical interfaces and interface units in RBFS, reflecting Layer 1 and Layer 2/3 operation.

Host Interfaces

Linux virtual ethernet (veth) interfaces connect an LXC container with the Linux host OS. In RBFS, a veth interface to the Linux bridge lxcbr0 is created by default. In virtual topologies, you can create additional veth interfaces and

Linux bridges. RBFS host interfaces represent veth interfaces in RBFS.
For example, if the container interface eth1 connects to the host interface vethXYZ123, ifp-0/0/1 can be bound to eth1 to represent it in RBFS. Host interfaces can be used like any other physical interface.

Memory Interfaces

Memory interfaces (memif) are virtual interfaces used for creating virtual topologies. They connect multiple containers running RBFS to each other. When configuring memif interfaces:

- Endpoints match on the memif ID, i.e. the memif ID needs to be the same on both ends.
- memif IDs need to be unique on the host.
- The memif interface name is locally significant only.
- One endpoint needs to be configured as a master, while the other one is configured as a slave.

Interface Numbering

RBFS interface numbers match the port numbers on the switch faceplate. An interface is named in the ifp-<chassis-ID>/<front-panel-block-number>/<port> format. For example, ifp-0/0/1.

- Chassis ID—always 0 for the currently supported platforms
- Front Panel Block—represents group of ports on the faceplate
- Port—matches the port number on switch faceplate

Virtual interfaces follow the same structure, for example, lo-0/0/1 or memif-0/0/1.

Logical interfaces are numbered: ifl-<Node ID>/<Chip ID>/<Port ID>/<Unit ID>, for example ifl-0/0/1/1.

Community Support for Interfaces

You can tag an interface address with a community or extended community. RBFS will create a direct route for each interface address. If a community or extended community is configured for an interface address, RBFS will add it to the direct route. Communities can be used in policies. For example, when redistributing direct routes, you can match these communities and define desired policy rules.

Unnumbered Interfaces

An unnumbered interface is a point-to-point interface that is not explicitly configured with a dedicated IP address and subnet. Instead, it borrows (or links to) from a loopback interface, and uses it as the source IP address for packets originating from the interface. The IP unnumbered interface can "borrow" the IP address from another interface that is already configured on the switch, thereby conserving network and address space.

The IP address of the unnumbered interface cannot be borrowed as it has no dedicated IP address. A logical interface can borrow IP address from a loopback interface, not vice versa.

Auto-negotiation

Auto-negotiation allows directly connected devices to automatically exchange speed and duplex mode information for the links. If auto-negotiation is enabled, ports can auto-negotiate the speed and duplex capabilities with other ports. The auto-negotiation can determine the best speed and duplex at which the ports can operate optimally.

> ℹ Port speed configuration and auto-negotiation are mutually exclusive.

RBFS supports auto-negotiation between ports in the following ways:

- 1G ports can negotiate with 10G ports.
- 40G ports can negotiate with 100G ports.

Auto-negotiation is not supported for the following combinations:

- 40G ports cannot negotiate with 1G ports, 10G ports, and 25G ports.
- 100G ports cannot negotiate with 1G ports, 10G ports and 25G ports.
- 1G port cannot negotiate with 25G port.

Interface Counters

Interface counters are statistics that network devices maintain for the traffic passing through the interfaces for all physical, logical, and LAG interfaces on a device. These counters provide information about the utilization and performance of the interface. Users can identify and troubleshoot issues such as congestion, errors, and performance bottlenecks by monitoring these counters.

RBFS Logical Interface counters and descriptions

Counters	Descriptions
Rx packets	The number of IP packets received by the logical interface.
Rx bytes	The number of bytes received by the logical interface.
Tx packets	The number of packets transmitted by the logical interface.
Tx bytes	The number of bytes transmitted by a logical interface.
IPv4 packets	The number of IPv4 packets processed by a logical interface.
IPv6 packets	The number of IPv6 packets processed by a logical interface.
MPLS packets	The number of MPLS packets processed by a logical interface.
Punt packets	The number of packets that are punted or forwarded to the CPU for further processing by a logical interface.
Drops packets	The number of packets that were dropped by a logical interface.
Rx Miss packets	The number of packets that were dropped by a logical interface.
Rx Error packets	The number of packets that were received with errors by the logical interface.
Rx No Buff packets	The number of packets that could not be received due to a lack of buffer space. When a logical interface receives a packet but does not have enough buffer space to store it, the packet is dropped.

Counters	Descriptions
Tx Error packets	The number of packets that could not be transmitted due to errors encountered during the transmission process.
Packet Statistics	
Ingress forwarded packets	The number of packets that are received by the device on one interface and then forwarded out on another interface.
Ingress forwarded bytes	The number of bytes that are received by a device on one interface and then forwarded out on another interface.
Ingress drop Packets	The number of packets that are dropped (discarded) by a network device upon arrival on one of its interfaces by a device.
Ingress drop bytes	Total number of bytes that have been dropped by an interface on incoming traffic.
Egress forwarded packets	Total number of packets that have been successfully forwarded by an interface on outgoing traffic.
Egress forwarded bytes	The number of bytes that have been forwarded by an interface in the egress direction.
Egress drop packets	The number of packets that have been dropped by an interface in the egress (outgoing) direction.
Egress drop bytes	The number of bytes that have been dropped by an interface in the egress direction.

RBFS physical interface counters and descriptions

Physical Interface Counters	Descriptions
VPP Statistics	
Rx packets	The number of IP packets received by the physical interface.
Rx bytes	The number of bytes received by the physical interface.

Physical Interface Counters	Descriptions
Tx packets	The number of packets transmitted by the physical interface.
Tx bytes	The number of bytes transmitted by the physical interface.
IPv4 packets	The number of IPv4 packets processed by the physical interface.
IPv6 packets	The number of IPv6 packets processed by the physical interface.
MPLS packets	The number of MPLS packets processed by a physical interface.
Punt packets	The number of packets that are punted or forwarded to the CPU for further processing by a physical interface.
Drops packets	The number of packets that were dropped by a physical interface.
Rx Miss packets	The number of packets that were dropped by a physical interface.
Rx Error packets	The number of packets that were received with errors by the physical interface.
Rx No Buff packets	The number of packets that could not be received due to a lack of buffer space. When a physical interface receives a packet but does not have enough buffer space to store it, the packet is dropped.
Tx Error packets	The number of packets that were dropped by a physical interface.
BCM Statistics	
inOctets	Number of octets received through the interface.
inUcastPkts	Number of unicast packets received through the interface.
inNonUcastPkts	Number of non-unicast packets received through the interface.

Physical Interface Counters	Descriptions
inErrors	The number of inbound packets that contained errors preventing them from being processed correctly.
outOctets	Number of octets sent through the interface.
ifHCInOctets	The number of octets (8-bit bytes) received by a network interface. It is a part of the SNMP MIB (Management Information Base) structure and is used to track high-capacity input octets counter used in the context of SNMP monitoring.
outUcastPkts	Number of unicast packets sent through the interface.
outNonUcastPkts	Number of non-unicast packets sent through the interface.
outErrors	The number of outbound packets that contained errors preventing them from being processed correctly.
etherStatsDropEvents	Number of events where packets were not delivered to the protocol stack because of resource limitations or other reasons. These dropped events can occur due to buffer overflows, congestion, or hardware limitations.
etherStatsMulticastPkts	The number of packets received by an interface that were addressed to a multicast MAC address.
etherStatsBroadcastPkts	The number of broadcast packets received on an Ethernet interface.
etherStatsUndersizePkts	The number of received packets that are smaller than the minimum allowed Ethernet frame size. Undersized packets can indicate various issues.
etherStatsFragments	The number of received packets that are fragments of IP datagrams.
etherStatsOversizePkts	The number of received packets that exceed the maximum Ethernet frame size.

Physical Interface Counters	Descriptions
etherStatsOctets	The total number of octets (bytes) of data transmitted and received on an Ethernet interface. This counter provides a measure of the total amount of data traffic on the interface.
etherStatsPkts	The number of packets transmitted or received by an Ethernet interface. This counter can provide insights into the traffic load and performance of the interface.
dot1dBasePortMtuExceededDiscards	The number of frames that were discarded at an interface as they exceeded the Maximum Transmission Unit (MTU) of the port. This typically happens when a frame is larger than the maximum size allowed on the interface and cannot be fragmented, so it is dropped.
etherStatsTXNoErrors	The number of Ethernet frames transmitted without any errors through the Ethernet interface. Each time a frame is successfully transmitted without encountering any errors, this counter is incremented.
etherStatsRXNoErrors	The number of Ethernet frames received without any errors through the Ethernet interface. Each time a frame is successfully transmitted without encountering any errors, this counter is incremented.
inMulticastPkts	The number of packets received by the interface that were addressed to a multicast address. These counters provide insights into the amount of multicast traffic being received by the interface.
outBroadcastPkts	The number of packets transmitted by the network interface as broadcast packets. These counters can provide insights into the amount of broadcast traffic generated by the interface.
outMulticastPkts	The number of packets transmitted by the interface as multicast packets. These counters can provide insights into the amount of multicast traffic generated by the interface.

Physical Interface Counters	Descriptions
outBroadcastPkts	The number of packets transmitted by the interface as broadcast packets. These counters can provide insights into the amount of broadcast traffic generated by the interface, which can be useful for network troubleshooting and monitoring network performance.
bcmReceivedUndersizePkts	The number of undersized packets received by a Broadcom device. Undersized packets are Ethernet frames that are smaller than the minimum allowed size. This counter provides insights into packet size issues.
bcmTransmittedUndersizePkts	The number of undersized packets sent by a Broadcom device. Undersized packets are Ethernet frames that are smaller than the minimum allowed size. This counter provides insights into packet size issues.
etherTxOversizePkts	The number of packets that exceed the maximum transmission unit (MTU) size allowed on an Ethernet interface.
etherStatsJabbers	The number of jabber frames received by an Ethernet interface. Jabber frames are Ethernet frames that exceed the maximum allowed frame size and contain data that extends beyond the maximum length.
etherStatsCRCAlignErrors	The number of frames received by an Ethernet interface that has a CRC (Cyclic Redundancy Check) error. Also, the frames are not an integral number of octets in length; that is, the frame length is not a multiple of 8 bits. CRC errors occur when the CRC checksum calculated by the receiving interface does not match the CRC checksum transmitted by the sending interface, indicating that the frame may have been corrupted during transmission

Physical Interface Counters	Descriptions
dot3StatsFCSErrors	The number of frames that have a Frame Check Sequence (FCS) error received by an Ethernet interface. The FCS is a field in the Ethernet frame that contains a checksum calculated based on the contents of the frame.
ifHCOutMulticastPkts	The number of outbound multicast packets on a network interface. The ifHCOutMulticastPkts object uses a 64-bit counter, allowing it to accommodate high-speed interfaces without wrapping around as quickly. It provides an accurate count of the outbound multicast packets on the interface.
ifHCOutBroadcastPckts	The ifHCOutBroadcastPkts counter provides the number of outbound broadcast packets on an interface. It is part of the IF-MIB (Interface MIB) and is an extension of the standard ifOutBroadcastPkts counter, providing a 64-bit counter for high-speed interfaces to avoid wrapping around quickly.

Path MTU Discovery

The Path MTU Discovery technique determines the maximum transmission unit between two hosts so that IP fragmentation can be avoided. By default, path MTU discovery is enabled in RBFS.

When RBFS receives MTU-violated packets, it will respond with the following ICMP error message:

ICMPv4 Message Types

The type field identifies the type of the message sent by the host. The type field contains more specific information about the error condition.

The table below lists the ICMPv4 message types.

Type	Description
3	Destination Unreachable. This alerts a source host of delivery problems encountered while trying to reach the destination.

Destination Unreachable uses the following code values to further describe the function of the ICMP message being sent.

Code	Description
4	Fragmentation Needed and Don’t-Fragment (DF) was Set. This message occurs when a router receives a packet that requires fragmentation, but the router has the DF DF flag turned on.

ICMPv6 Message Types

The type field identifies the type of the message sent by the host. The type field contains more specific information about the error condition.

The table below lists the ICMPv6 message type.

Type	Description
2	Packet Too Big. A Packet Too Big MUST be sent by a router in response to a packet that it cannot forward because the packet is larger than the MTU of the outgoing link.

Code	Description
0	No code

You can change this behavior by enabling fragmentation. For more information about enabling hostpath fragmentation, see the section "Enabling Hostpath Fragmentation" below.

All outgoing packets are validated against the configured MTU on the egress path.

- If MTU is violated and MTU-profile action is drop, then packets are dropped in hardware.

- If MTU is violated and MTU-profile action is redirect-to-cpu, a 20MB policer is used to protect the CPU-port from overwhelming MTU-violated traffic, and packets are sent to the CPU port.
- When fragmentation is enabled, one of the following operations takes place.

  If the DF bit is not set in the received packet (only for IPv4), the packets are fragmented and sent to the outgoing port.

  If the DF bit is set in a packet, it drops the packet and sends an ICMP error message back to the source.
- When fragmentation is disabled, packets are dropped and ICMP error messages are sent to the source.

For information about configuring the MTU profile, see MTU Profile Configuration.

IP Fragmentation

If the maximum transmission unit (MTU) of an outgoing interface is less than the original packet which needs to be routed, the packet needs to be fragmented.

RBFS supports IP fragmentation on the QMX and QAX platforms but not on the Q2C platform. However, currently, there is no support for IP fragmentation in the QMX, QAX, or Q2C hardware. Due to this limitation, on the QMX and QAX platforms, the packets are sent to the CPU, and the fragmentation is handled by the CPU therefore the rate for these packets is significantly reduced.

If the packet that needs to be fragmented and the Do-Not-Fragment (DF) bit is specified, then the device is going to send an ICMP Error code "fragmentation needed and DF set" to the source.

By default, IPv6 fragmentation is handled at source. When the transit device receives an MTU-violated packet, it sends a "Packet Too Big" ICMPv6 message that it cannot forward the packet because it is larger than the MTU of the outgoing link.

Guidelines and Limitations of IP Fragmentation

The following guidelines and limitations apply to IP Fragmentation:

- If a packet is larger than the negotiated subscriber MTU size, it will be fragmented (on the QMX platform); whereas on the Q2C platform, such a packet will be dropped. You can control the fragmentation on the Q2C platform by configuring the set forwarding-options fragmentation ipv4 state CPU command. For more information about configuring fragmentation, see "2.2.2. Enabling Hostpath Fragmentation."

- The fragmented packets do not go over the regular QoS path in the egress pipeline.
- There will be no ICMP error message sent in response to MTU-violated multicast packets.

MTU Profile

The Maximum Transmission Unit (MTU) is the size of the packet that is allowed in the network. In the new generation silicon like Broadcom Qumran2C (Q2C), resources are conserved by creating profiles of the resources, and multiple entities like IFP, IFL and L3 interfaces utilize these profiles. To better manage MTU resources and platform capabilities, RBFS supports configuring MTU profiles and attaching these profiles to the attachment points.

Attachment Points

The MTU profiles are attached to the interface entities like physical (IFP), logical (IFL) and L3 interfaces. RBFS supports the below attachment points for MTU profiles:

- Port-level
- L3 interface level (IPv4 and IPv6)
- PPPoE subscriber level (L2 IFL)

MTU Size

A user-configured MTU size can range from 64 to 9216 in RBFS.

For MTU profiles of type "pppoe", users should provide L3 MTU size (IPv4/IPv6 headers).

MTU Type

An MTU type specifies the attachment point of the MTU profile. The MTU types supported are as follows:

- **physical**: When checking MTU, the entire packet size is considered.
- **ipv4**: MTU check is based on IPv4 headers.

- **ipv6**: MTU check is based on IPv6 headers.
- **ip**: MTU profile of type IP.
- **pppoe**: The MTU profile is applied to the PPPoE subscriber interface and the user is required to provide the L3 MTU size. Based on its best match algorithm, the Subscriber Management service associates these profiles with PPPoE subscribers.

> - MTU profiles for L3 logical interfaces must be explicitly configured, and if not configured, no default MTU size is set for IPv4 and IPv6.
> - On the Q2C platform, L3 interfaces can only be configured with IPv4 MTU profile or IPv6 MTU profile, but not both. However, with the type "ip" MTU profile, you can configure MTU for both IPv4 and IPv6 traffic with a common MTU size.
> - On the QAX platform, only physical MTU profiles are supported.

MTU Action

The MTU action defines the action to be taken when the MTU check fails. Currently, RBFS supports “drop” as an action.

MTU Profile Limitations

The following limitations apply to the MTU profile:

- There is a limit to how many MTUs can be used by each hardware.

  On the Q2C platform, the limit is as follows:

  Maximum number of MTU profiles: 8

  Maximum number of L3 MTU profiles: 3 (MTU type: IP/IPv4/IPv6)

  Maximum number of PPPoE MTU profiles: 6 (including the default PPPoE profile)

  Maximum number of physical MTU profiles: 7

  On the QAX platform, the limit is as follows:

  Maximum number of physical MTU profiles: 1

Supported Platforms

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

Guidelines & Limitations

QAX-based Platforms

An additional restriction applies to ports on QAX-based platforms: because of hardware design, physical ports are grouped into quads (groups of 4, also known as port groups). Each quad must have the same physical parameters: speed, link-training, duplex.

The following tables are provided for easy identification of ports that need to have the same physical settings:

Edgecore 7316-26XB Port Groups:

Port	Speed	Duplex	Port Group
ifp-0/0/0	100G	Full	0
ifp-0/0/1	100G	Full	1
ifp-0/1/0	10G	Full	2
ifp-0/1/1	10G	Full	
ifp-0/1/2	10G	Full	
ifp-0/1/3	10G	Full	
ifp-0/1/4	10G	Full	3
ifp-0/1/5	10G	Full	
ifp-0/1/6	10G	Full	
ifp-0/1/7	10G	Full	

Port	Speed	Duplex	Port Group
ifp-0/1/8	10G	Full	4
ifp-0/1/9	10G	Full	
ifp-0/1/10	10G	Full	
ifp-0/1/11	10G	Full	
ifp-0/1/12	10G	Full	5
ifp-0/1/13	10G	Full	
ifp-0/1/14	10G	Full	
ifp-0/1/15	10G	Full	
ifp-0/1/16	25G	Full	6
ifp-0/1/17	25G	Full	
ifp-0/1/18	25G	Full	
ifp-0/1/19	25G	Full	
ifp-0/1/20	25G	Full	7
ifp-0/1/21	25G	Full	
ifp-0/1/22	25G	Full	
ifp-0/1/23	25G	Full	

UfiSpace S9500-22XST Port Groups:

Port	Speed	Duplex	Port Group
ifp-0/0/0	10G	Full	8
ifp-0/0/1	10G	Full	
ifp-0/0/2	10G	Full	
ifp-0/0/3	10G	Full	
ifp-0/0/4	10G	Full	4
ifp-0/0/5	10G	Full	
ifp-0/0/6	10G	Full	
ifp-0/0/7	10G	Full	

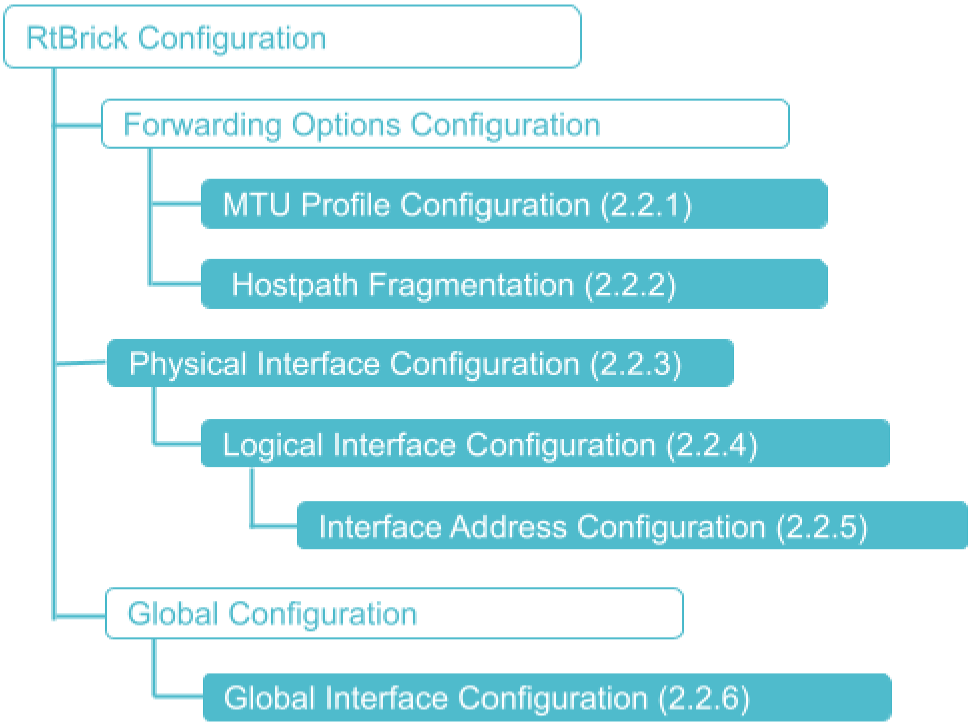
Port	Speed	Duplex	Port Group
ifp-0/0/8	10G	Full	11
ifp-0/0/9	10G	Full	
ifp-0/0/10	10G	Full	
ifp-0/0/11	10G	Full	
ifp-0/0/12	25G	Full	3
ifp-0/0/13	25G	Full	
ifp-0/0/14	25G	Full	
ifp-0/0/15	25G	Full	
ifp-0/0/16	25G	Full	2
ifp-0/0/17	25G	Full	
ifp-0/0/18	25G	Full	
ifp-0/0/19	25G	Full	
ifp-0/0/20	100G	Full	0
ifp-0/0/21	100G	Full	1

A PHY Quad can be associated with network interface (NIF) ports of identical type only. For example, a quad cannot be a mix of XLGE and XE ports. An exception is GE and XE ports which can coexist in the same quad. This means all the ports in a port group should have the same physical interface configuration (that is, speed/duplex/link-training). Ports in a port group are only allowed to support 1G and 10G speeds; any other combination is not allowed. If a port within a port group is misconfigured, then it would require changing the speeds/interface type of all ports within the port group to a different type and then back into the original type.

5.1.2. Interfaces Configuration

Configuration Hierarchy

The diagram illustrates the interface configuration hierarchy.


RtBrick Configuration
Forwarding Options Configuration
MTU Profile Configuration (2.2.1)
Hostpath Fragmentation (2.2.2)
Physical Interface Configuration (2.2.3)
Logical Interface Configuration (2.2.4)
Interface Address Configuration (2.2.5)
Global Configuration
Global Interface Configuration (2.2.6)

Configuration Syntax and Commands

The following sections describe the interface configuration syntax and commands.

MTU Profile Configuration

This section describes how to configure MTU profiles.

Syntax:

set forwarding-options mtu-profile <attribute> <value>

Attribute	Description
mtu-profile <mtu-profile-name>	MTU profile name
mtu-size <mtu-size>	MTU size. Range: 64 to 9216 bytes

Attribute	Description
mtu-type <mtu-type>	Specify the MTU type: • **physical**: Port based MTU profile • **pppoe**: subscriber IFL-based MTU profile for L2TP and PPPoE. This MTU profile is used by PPPoE subscribers to set the default MTU size of 1492. A configured size of 1492 bytes limits the size of the IPv4 or IPv6 header plus payload. • **ipv4**: MTU profile of type IPv4. Only IPv4 traffic on the logical interface will be impacted. • **ipv6**: MTU profile of type IPv6. Only IPv6 traffic on the logical interface will be impacted. • **ip**: MTU profile of type IP. Both IPv4 and IPv6 traffic on the logical interface will be impacted.
action <mtu-action>	Specify the MTU action. The following options are supported: **drop**: This indicates that when the MTU check fails, the action "drop" is performed. **redirect-to-cpu**: This is an action of redirecting packets to the CPU in a traffic behavior. A redirect-to-cpu action must be configured for fragmentation to occur.

Example 1: Configuration of the MTU Profile for the Physical Port

```
{
  "ietf-restconf:data": {
    "rtbrick-config:forwarding-options": {
      "mtu-profile": [
        {
          "mtu-profile-name": "portMtu",
          "size": 5000,
          "type": "physical",
          "action": "redirect-to-cpu"
        }
      ]
    }
  }
}
```

Example 2: MTU Profile Configuration of Type IPv4

```
{
  "ietf-restconf:data": {
    "rtbrick-config:forwarding-options": {
      "mtu-profile": [
        {
          "mtu-profile-name": "ipv4Mtu",
          "size": 1300,
          "type": "ipv4",
          "action": "redirect-to-cpu"
        }
      ]
    }
  }
}
```

Example 3: MTU Profile Configuration of Type IPv6

```
{
  "ietf-restconf:data": {
    "rtbrick-config:forwarding-options": {
      "mtu-profile": [
        {
          "mtu-profile-name": "ipv6Mtu",
          "size": 1400,
          "type": "ipv6",
          "action": "redirect-to-cpu"
        }
      ]
    }
  }
}
```

Example 4: Configuration of the MTU Profile for PPPoE

```
{
  "ietf-restconf:data": {
    "rtbrick-config:forwarding-options": {
      "mtu-profile": [
        {
          "mtu-profile-name": "pppoeMtu",
          "size": 1492,
          "type": "pppoe",
          "action": "redirect-to-cpu"
        }
      ]
    }
  }
}
```

Enabling Hostpath Fragmentation

This section describes how to enable or disable fragmentation by CPU. It is necessary to configure MTU profile action "redirect-to-cpu" so that fragmentation takes place. By default, fragmentation is disabled.

Syntax:

set forwarding-options fragmentation ipv4 state <value>

Attribute	Description
disabled | cpu	Enables fragmentation of IPv4 packets. There are two options: disabled—Fragmentation is disabled. It is the default setting. cpu—Fragmentation is performed by CPU.

Example: Configuration of Hostpath Fragmentation

```
{
  "ietf-restconf:data": {
    "rtbrick-config:forwarding-options": {
      "fragmentation": {
        "ipv4": {
          "state": "cpu"
        }
      }
    }
  }
}
```

Physical Interface Configuration

This section describes configuration options at the physical interface (IFP) level.

Syntax:

set interface <interface-name> <attribute> <value>

Attribute	Description
<interface-name>	Name of the interface. Example: ifp-0/0/1.
admin-status <down|up>	Administrative state of the interface.

Attribute	Description
auto-negotiation true	Enable auto-negotiation. Note: To disable auto-negotiation, use the delete form of the command. NOTE: Port speed configuration and auto-negotiation are mutually exclusive.
class-of-service <profile-name>	Apply class-of-service profile name.
description	Configure physical interface description.
host-if <container-interface>	Configure a host interface. For example, if the container interface eth1 connects to the host interface vethXYZ123, use this command option to bound hostif-0/0/1 to eth1. Please note the Linux virtual ethernet (veth) interface needs to be created separately. It cannot be created via RBFS configuration.
forward-error-correction <fec-type>	Configure Forward Error Correction (FEC) on the physical interface. FEC allows you to send the required information to correct errors through the link along with the payload data. A benefit of "forward" in FEC is that errors detected at the receiver do not need to be retransmitted. Currently, the supported FEC types are: base-r, rsfec, none. NOTE: rsfec is the only FEC supported for 100G on the QAX platform.
link-training true	Enable link training. Note: To disable link training, use the delete form of the command.
master <true|false>	Memif role, master or slave, applicable only to memif interface. One end needs to be configured as master, and the other one as slave.
memif-id <id>	Configure memif ID, applicable only to memif interface. Needs to match on both ends.
mtu-profile <mtu-profile-name>	Attach MTU profile to a physical interface. This is a mandatory attribute.

Attribute	Description
mru <size>	Maximum receive unit size on the physical interface.
speed <speed>	Configure speed mode for the interface. Port speed refers to the maximum amount of data transmitted. The speed value is specified in Gigabits per second (Gbps). Currently, RBFS supports 10G and 100G ports, and you can make the following changes: • 100G port speed can be changed to 40G • 10G port speed can be changed to 1G

Example 1: Physical Interface Configuration

```
{
    "rtbrick-config:interface": [
      {
        "name": "ifp-0/0/1",
        "description": "Link to leaf1",
        "speed": "10G",
        "mtu-profile": "portMtu",
        "mru": 5000
      }
    ]
}
```

Example 2: Memory Interface Configuration

A End:

```
{
    "rtbrick-config:interface": [
      {
        "name": "ifp-0/0/1",
        "description": "Master",
        "memif-id": 11,
        "master": "true",
      }
    ]
}
```

B End:

```
{
    "rtbrick-config:interface": [
      {
        "name": "ifp-0/0/1",
        "description": "Slave",
        "memif-id": 11,
        "master": "false",
      }
    ]
  }
```

Example 3: Host Interface Configuration

```
{
    "rtbrick-config:interface": [
        "name": "ifp-0/0/1",
        "description": "Represents eth1 as ifp-0/0/1 in RBFS",
        "host-if": "eth1",
    ]
}
```

Example 4: MRU Configuration for Physical Interface

```
{
    "rtbrick-config:interface": [
      {
        "name": "ifp-0/0/7",
        "mru": 5000
      }
    ]
  }
```

Example 5: FEC Configuration for Physical Interface

```
{
    "rtbrick-config:interface": [
      {
        "name": "ifp-0/0/40",
        "forward-error-correction": "base-r"
      }
    ]
  }
```

Logical Interface Configuration

This section describes configuration options at the logical interface (IFL) level.

Syntax:

set interface <interface-name> **unit** <unit-id> <attribute> <value>

Attribute	Description
unit <unit-id>	Create a logical interface (also referred to as a sub-interface) under the physical interface.
admin-status <down|up>	Administrative state of the logical interface.
class-of-service <profile-name>	Apply class-of-service profile name.
description <description>	Description of the logical interface.
inner-vlan <inner-vlan-id>	Inner VLAN ID.
instance <instance>	Assign the logical interface to an instance.
ipv4-admin-status <down|up>	Enable or disable IPv4.
ipv4-mtu-profile <ipv4-mtu-profile>	Attach IPv4 MTU profile to an L3 interface.
ipv6-admin-status <down|up>	Enable or disable IPv6.
ipv6-mtu-profile <ipv6-mtu-profile>	Attach IPv6 MTU profile to an L3 interface.
ip-mtu-profile <ip-mtu-profile>	Attach IP MTU profile to an L3 interface.
mpls-admin-status <down|up>	Enable or disable MPLS.
mpls-mtu <mpls-mtu-size>	MPLS maximum transmission unit size.
neighbor <ipv4|ipv6> <ip-address> mac <mac-address>	Configure a static IPv4 or IPv6 neighbor.
unnumbered interface <loopback-interface-name>	Configure an un-numbered interface.
vlan <outer-vlan-id>	Outer VLAN ID.

Example 1: Logical Interface Configuration with IPv4 MTU Profile

```
 {
   "rtbrick-config:interface": [
      {
        "name": "ifp-0/0/1",
        "unit": [
          {
            "unit-id": 1,
            "description": "VLAN 101",
            "instance": "default",
            "ipv4-mtu-profile": "ipv4Mtu"

          }
        ]
      }
   ]
 }
```

Example 2: Logical Interface Configuration with IPv6 MTU Profile

```
 {
   "rtbrick-config:interface": [
      {
        "name": "ifp-0/0/1",
        "unit": [
          {
            "unit-id": 1,
            "description": "VLAN 101",
            "instance": "default",
            "ipv6-mtu-profile": "ipv6Mtu"

          }
        ]
      }
   ]
}
```

Example 3: Logical Interface Configuration with IP MTU Profile

```
 {
   "rtbrick-config:interface": [
      {
        "name": "ifp-0/0/1",
        "unit": [
          {
            "unit-id": 1,
            "description": "VLAN 101",
            "instance": "default",
            "ip-mtu-profile": "ipMtu"

          }
        ]
      }
   ]
}
```

Interface Address Configuration

This section describes how to configure interface IP addresses.

Syntax:

set interface <interface-name> **unit** <unit-id> **address** <afi> <attribute> <value>

Attribute	Description
<afi>	Address family identifier (AFI). Supported values: ipv4 and ipv6
<prefix4|prefix6>	Assign IPv4 or IPv6 address to the interface unit.
community <community-value>	Configure list of communities associated with the address.
extended-community <community-value>	Configure list of extended communities associated with the address.
label <label-value>	Configure label associated with the address. Supported MPLS label values are 0 - 1048575. The reserved MPLS label range is 0 - 15. In RBFS, BGP uses the label range 20000 - 100000. It is recommended to assign label values outside of these reserved ranges to avoid conflicts.
secondary true	Enable a secondary IPv4 address. Note: To disable the secondary IP configuration, use the delete form of the command.

> Broadcast and network IP addresses cannot be configured on logical (IFL) interfaces.

Example: Interface Address Configuration

```
{
    "rtbrick-config:interface": [
      {
        "name": "lo-0/0/1",
        "unit": [
          {
            "unit-id": 1,
            "address": {
              "ipv4": [
                {
                  "prefix4": "198.51.100.103/24",
                  "label": 12346
                }
              ]
            }
          }
        ]
      }
    ]
  }
```

Global Interface Configuration

This section describes a configuration option applied globally to all interfaces.

Syntax:

set global interface all <attribute> <value>

Attribute	Description
admin-status <up|down>	Configure state of the interface.

- The interface level enable/disable command has higher precedence than the global interface enable/disable command.
- You can disable all unused physical interfaces.
- Before executing the global interface disable all command ensure that all physical interfaces are in the link Up state.

Example: Enabling or Disabling all Interfaces

```
{
  "ietf-restconf:data": {
    "rtbrick-config:global": {
      "interface": {
        "all": {
          "admin-status": "down"
        }
      }
    }
  }
}
```

5.1.3. Interfaces Operational Commands

Interface Show Commands

The interface show commands provide detailed information about the status and parameters of RBFS interfaces.

Interface Summary Commands

Syntax:

show interface <option>

Option	Description
summary	Displays a summary of all interfaces including physical, logical, and address information.
<interface-name>	Displays a summary of an interface including physical, logical, and address information.
physical	Displays all physical interface including loopback, cpu and recycle ports.
logical	Displays all logical interfaces for all instances.
logical <instance-name>	Displays all logical interfaces for the given instance.
address	Displays all IPv4 and IPv6 addresses for all instances.
address <instance-name>	Displays all IPv4 and IPv6 addresses for the given instance.

Example 1: Summary Output for All Interfaces

```
supervisor@rtbrick>LEAF01: op> show interface summary
Interface                Admin    Link     Oper          IPv4 Address             IPv6 Address
ifp-0/0/1                Up       Down     Down
ifp-0/0/2                Up       Down     Down
ifp-0/0/3                Up       Down     Down
ifp-0/0/4                Up       Up       Up
ifp-0/0/5                Up       Down     Down
ifp-0/0/6                Up       Down     Down
ifp-0/0/7                Up       Down     Down
ifp-0/0/8                Up       Down     Down
ifp-0/0/9                Up       Down     Down
ifp-0/0/10               Up       Up       Up
ifl-0/0/10/100           Up       Up       Up            198.51.100.22/24         2001:db8:0:100::/32
ifl-0/0/10/200           Up       Up       Up            198.51.100.32/24         2001:db8:0:10::/32
ifl-0/0/10/300           Up       Up       Up            -                        2001:db8:0:160::/32
ifp-0/0/11               Up       Down     Down
ifp-0/0/12               Up       Down     Down
ifp-0/0/13               Up       Down     Down
ifp-0/0/14               Up       Down     Down
ifp-0/0/15               Up       Down     Down
ifp-0/0/16               Up       Down     Down
```

```
ifp-0/0/17               Up       Down     Down
ifp-0/0/18               Up       Down     Down
ifp-0/0/19               Up       Down     Down
ifp-0/0/20               Up       Down     Down
ifp-0/0/21               Up       Down     Down
ifp-0/0/22               Up       Down     Down
ifp-0/0/23               Up       Down     Down
ifp-0/0/24               Up       Down     Down
ifp-0/0/25               Up       Down     Down
ifp-0/0/26               Up       Down     Down
ifp-0/0/27               Up       Up       Up
ifp-0/0/28               Up       Down     Down
ifp-0/0/29               Up       Down     Down
ifp-0/0/30               Up       Down     Down
ifp-0/0/31               Up       Down     Down
ifp-0/0/32               Up       Down     Down
ifp-0/0/33               Up       Down     Down
ifp-0/0/34               Up       Down     Down
ifp-0/0/35               Up       Down     Down
ifp-0/0/36               Up       Down     Down
ifp-0/0/37               Up       Down     Down
ifp-0/0/38               Up       Down     Down
ifp-0/0/39               Up       Down     Down
ifp-0/0/40               Up       Down     Down
ifp-0/0/41               Up       Down     Down
ifp-0/0/42               Up       Down     Down
ifp-0/0/43               Up       Down     Down
ifp-0/0/44               Up       Down     Down
ifp-0/0/45               Up       Down     Down
ifp-0/0/46               Up       Down     Down
ifp-0/0/47               Up       Down     Down
ifp-0/0/48               Up       Down     Down
ifp-0/0/49               Up       Down     Down
ifp-0/0/50               Up       Down     Down
ifp-0/0/51               Up       Down     Down
ifp-0/0/52               Up       Up       Up
ifp-0/0/53               Up       Up       Up
ifp-0/0/54               Up       Down     Down
cpu-0/0/200              Up       Up       Up
cpu-0/0/201              Up       Down     Down
cpu-0/0/202              Up       Down     Down
cpu-0/0/203              Up       Down     Down
recycle-0/0/75           Up       Up       Up
recycle-0/0/75/0         Up       Up       Up
recycle-0/0/76           Up       Up       Up
recycle-0/0/76/0         Up       Up       Up
```

Example 2: Summary Output for One Physical Interface

```
supervisor@rtbrick>LEAF01: op> show interface ifp-0/0/10
Interface                Admin    Link     Oper          IPv4 Address             IPv6 Address
ifp-0/0/10               Up       Up       Up
  ifl-0/0/10/100         Up       Up       Up            198.51.100.22/24               2001:db8:0:100::/32
  ifl-0/0/10/200         Up       Up       Up            198.51.100.32/24               2001:db8:0:10::/32
  ifl-0/0/10/300         Up       Up       Up            -                        2001:db8:0:160::/32
  ifl-0/0/10/1000        Up       Up       Up            -                        2001:db8:0:33::/32
```

Example 3: List of All Physical Interfaces

```
supervisor@rtbrick>LEAF01: op> show interface physical
Interface        Admin   Link    Oper    MAC Address          Speed  Duplex  Uptime
lo-0/0/1         Up      Up      Up      80:a2:35:a0:00:01    -      -       Thu Nov 19 10:41:06 GMT +0000
2020
ifp-0/0/1        Up      Down    Down    80:a2:35:ee:a8:01    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/2        Up      Down    Down    80:a2:35:ee:a8:02    10G    Full    Mon Nov 16 11:24:09 GMT +0000
```

```
2020
ifp-0/0/3        Up      Down    Down    80:a2:35:ee:a8:03    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/4        Up      Up      Up      80:a2:35:ee:a8:04    10G    Full    Thu Nov 19 10:05:02 GMT +0000
2020
ifp-0/0/5        Up      Down    Down    80:a2:35:ee:a8:05    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/6        Up      Down    Down    80:a2:35:ee:a8:06    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/7        Up      Down    Down    80:a2:35:ee:a8:07    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/8        Up      Down    Down    80:a2:35:ee:a8:08    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/9        Up      Down    Down    80:a2:35:ee:a8:09    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/10       Up      Up      Up      80:a2:35:ee:a8:0a    10G    Full    Fri Nov 20 00:59:12 GMT +0000
2020
ifp-0/0/11       Up      Down    Down    80:a2:35:ee:a8:0b    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/12       Up      Down    Down    80:a2:35:ee:a8:0c    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/13       Up      Down    Down    80:a2:35:ee:a8:0d    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/14       Up      Down    Down    80:a2:35:ee:a8:0e    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/15       Up      Down    Down    80:a2:35:ee:a8:0f    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/16       Up      Down    Down    80:a2:35:ee:a8:10    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/17       Up      Down    Down    80:a2:35:ee:a8:11    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/18       Up      Down    Down    80:a2:35:ee:a8:12    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/19       Up      Down    Down    80:a2:35:ee:a8:13    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/20       Up      Down    Down    80:a2:35:ee:a8:14    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/21       Up      Down    Down    80:a2:35:ee:a8:15    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/22       Up      Down    Down    80:a2:35:ee:a8:16    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/23       Up      Down    Down    80:a2:35:ee:a8:17    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/24       Up      Down    Down    80:a2:35:ee:a8:18    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/25       Up      Down    Down    80:a2:35:ee:a8:19    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/26       Up      Down    Down    80:a2:35:ee:a8:1a    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/27       Up      Up      Up      80:a2:35:ee:a8:1b    10G    Full    Fri Nov 20 00:59:11 GMT +0000
2020
ifp-0/0/28       Up      Down    Down    80:a2:35:ee:a8:1c    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/29       Up      Down    Down    80:a2:35:ee:a8:1d    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/30       Up      Down    Down    80:a2:35:ee:a8:1e    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/31       Up      Down    Down    80:a2:35:ee:a8:1f    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/32       Up      Down    Down    80:a2:35:ee:a8:20    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/33       Up      Down    Down    80:a2:35:ee:a8:21    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/34       Up      Down    Down    80:a2:35:ee:a8:22    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/35       Up      Down    Down    80:a2:35:ee:a8:23    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/36       Up      Down    Down    80:a2:35:ee:a8:24    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/37       Up      Down    Down    80:a2:35:ee:a8:25    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/38       Up      Down    Down    80:a2:35:ee:a8:26    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/39       Up      Down    Down    80:a2:35:ee:a8:27    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/40       Up      Down    Down    80:a2:35:ee:a8:28    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
```

```
ifp-0/0/41       Up       Down    Down    80:a2:35:ee:a8:29    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/42       Up       Down    Down    80:a2:35:ee:a8:2a    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/43       Up       Down    Down    80:a2:35:ee:a8:2b    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/44       Up       Down    Down    80:a2:35:ee:a8:2c    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/45       Up       Down    Down    80:a2:35:ee:a8:2d    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/46       Up       Down    Down    80:a2:35:ee:a8:2e    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/47       Up       Down    Down    80:a2:35:ee:a8:2f    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/48       Up       Down    Down    80:a2:35:ee:a8:30    10G    Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/49       Up       Down    Down    80:a2:35:ee:a8:31    100G   Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/50       Up       Down    Down    80:a2:35:ee:a8:35    100G   Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/51       Up       Down    Down    80:a2:35:ee:a8:39    100G   Full    Mon Nov 16 11:24:09 GMT +0000
2020
ifp-0/0/52       Up       Up      Up      80:a2:35:ee:a8:3d    100G   Full    Tue Nov 17 14:10:46 GMT +0000
2020
ifp-0/0/53       Up       Up      Up      80:a2:35:ee:a8:41    100G   Full    Fri Nov 20 00:59:12 GMT +0000
2020
ifp-0/0/54       Up       Down    Down    80:a2:35:ee:a8:45    100G   Full    Mon Nov 16 11:24:09 GMT +0000
2020
cpu-0/0/200      Up       Up      Up      80:a2:35:ee:a8:c8    100G   Full    Mon Nov 16 11:24:11 GMT +0000
2020
cpu-0/0/201      Up       Down    Down    80:a2:35:ee:a8:c9    100G   Full    Mon Nov 16 11:24:09 GMT +0000
2020
cpu-0/0/202      Up       Down    Down    80:a2:35:ee:a8:ca    100G   Full    Mon Nov 16 11:24:09 GMT +0000
2020
cpu-0/0/203      Up       Down    Down    80:a2:35:ee:a8:cb    100G   Full    Mon Nov 16 11:24:09 GMT +0000
2020
recycle-0/0/75   Up       Up      Up      80:a2:35:ee:a8:4b    100G   Full    Mon Nov 16 11:24:11 GMT +0000
2020
recycle-0/0/76   Up       Up      Up      80:a2:35:ee:a8:4c    100G   Full    Mon Nov 16 11:24:11 GMT +0000
2020
```

Example 4: List of All Logical Interfaces for All Instances

```
supervisor@rtbrick>LEAF01: op> show interface logical
Interface              Instance            Admin   Link    Oper    Outer VLAN   Inner VLAN  IPv4 Status,MTU
IPv6 Status,MTU
ifl-0/0/10/100         default             Up      Up      Up      -            -           Up,1500
Up,1500
ifl-0/0/10/200         default             Up      Up      Up      200          -           Up,1500
Up,1500
ifl-0/0/10/300         default             Up      Up      Up      300          -           Up,1500
Up,1500
```

Example 5: List of Logical Interfaces for an Instance

```
supervisor@rtbrick: op> show interface logical instance default
Interface              Instance            Admin   Link    Oper    Outer VLAN   Inner VLAN  IPv4 Status,MTU
IPv6 Status,MTU
ifl-0/0/10/100         default             Up      Up      Up      -            -           Up,1500
Up,1500
ifl-0/0/10/200         default             Up      Up      Up      200          -           Up,1500
Up,1500
ifl-0/0/10/300         default             Up      Up      Up      300          -           Up,1500
Up,1500
```

Example 6: List of All Interface Addresses

```
supervisor@rtbrick: op> show interface address
Interface              Instance         IPv4 Address        IPv4 Primary   IPv6 Address
  ifl-0/0/10/100       default        198.51.100.22/24      True          2001:db8:0:100::/32
  ifl-0/0/10/200       default        198.51.100.32/24       True         2001:db8:0:10::/32
  ifl-0/0/10/300       default             -                              2001:db8:0:160::/32
```

Interface Details Commands

Syntax:

show interface <option> **detail**

Option	Description
detail	Without any additional option, displays detailed information for all interfaces.
<interface-name> detail	Displays detailed information for an interface.

Example 7: Detailed Information for a Physical Interface and Its Logical Interfaces

```
supervisor@rtbrick: op> show interface ifp-0/0/10 detail
Interface:ifp-0/0/10
  Admin/Link/Operational status: Up/Up/Up
  MRU: 10000
  MTU: 1514
  Interface type: hostif
  Interface index: 8198
  MAC: 7a:4a:14:c0:00:04
  Uptime:  Thu Apr 11 05:57:42 GMT +0000 2024
  Flap count:  2
  Description: Physical interface #4 from node 0, chip 0
  Interface: ifp-0/0/10, Instance: default
    Admin/Link/Operational status: Up/Up/Up
    IPv4/IPv6/MPLS Status: Up/Up/Up
    IPv4/IPv6/MPLS MTU: 1500/1500/1500
    Interface type: Logical Sub interface
    Interface index: 532486
    MAC: 7a:4a:14:c0:00:04
    IPv4 Address             IPv6 Address
    192.0.2.0/24             2001:DB8::/64
```

MTU Profile Show Command

Syntax:

show mtu profile <option>

Option	Description
-	Without any additional option, displays detailed information for all MTU profiles.
profile-name <mtu-profile-name>	MTU Profile Name

Example 8: Detailed Information About the MTU Profiles

```
supervisor@rtbrick>LEAF01: op> show mtu profile
Profile Name               Type         Size   Action
__default_pppoe__          pppoe        1492   drop
l3IpMtu                    ipv4         1300   drop
l3Ipv6Mtu                  ipv6         1300   drop
portMtu                    physical     1300   drop
portM2                     physical     1400   drop
portM5                     physical     1430   drop
supervisor@rtbrick>LEAF01: op>
```

Example 9: Display Information About the Specified MTU Profile

```
supervisor@rtbrick>LEAF01: op> show mtu profile profile-name l3IpMtu
Profile Name               Type         Size   Action
l3IpMtu                    ipv4         1300   drop
supervisor@rtbrick>LEAF01: op>
```

Interface Statistics Commands

Syntax:

show interface <option> **statistics**

Option	Description
statistics	Without any additional option, displays statistics information for all interfaces.
<interface-name> statistics	Displays statistics information for an interface.

Example 10: Statistics Information for a Physical Interface and Its Logical Interfaces

```
supervisor@rtbrick>LEAF01: op> show interface ifp-0/0/10 statistics
Interface:  ifp-0/0/10
   Counter               Direction   Unit       Rx         Rx Diff    Rx Rate    Tx         Tx Diff   Tx Rate
   IPv4                  -           Packets    -          -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
```

```
    IPv6                  -           Packets    -          -          -          -          -          -
                                      Bytes      -          -          -          -          -          -
    MPLS                  -           Packets    -          -          -          -          -          -
                                      Bytes      -          -          -          -          -          -
    Punt                  -           Packets    -          -          -          -          -          -
                                      Bytes      -          -          -          -          -          -
    Miss                  RX          Packets    -          -          -          -          -          -
                                      Bytes      -          -          -          -          -          -
    Drops                 -           Packets    4995       -          -          -          -          -
                                      Bytes      -          -          -          -          -          -
    Error                 RX          Packets    -          -          -          -          -          -
                                      Bytes      -          -          -          -          -          -
    Error                 TX          Packets    47         -          -          -          -          -
                                      Bytes      -          -          -          -          -          -
    No Buff               RX          Packets    -          -          -          -          -          -
                                      Bytes      -          -          -          -          -          -
    Traffic Statistics    -           Packets    4995       -          -          68492      -          -
                                      Bytes      489510     -          -          5869876    -          -
    Unicast Statistics    -           Packets    -          -          -          -          -          -
                                      Bytes      -          -          -          -          -          -
    Broadcast Statistics  -           Packets    -          -          -          -          -          -
                                      Bytes      -          -          -          -          -          -
    Multicast Statistics  -           Packets    -          -          -          -          -          -
                                      Bytes      -          -          -          -          -          -
    Bcm Statistics:
    inOctets:                                 511632
    inUcastPkts:                              0
    inNonUcastPkts:                           5016
    inErrors:                                 0
    inUnknownProtos:                          0
    outOctets:                                6236484
    outUcastPkts:                             0
    outNonUcastPkts:                          68492
    outErrors:                                0
    etherStatsDropEvents:                     0
    etherStatsMulticastPkts:                  67718
    etherStatsBroadcastPkts:                  5790
    etherStatsUndersizePkts:                  0
    etherStatsFragments:                      0
    etherStatsOversizePkts:                   0
    etherStatsOctets:                         6748116
    etherStatsPkts:                           73508
    etherStatsCollisions:                     0
    etherStatsTXNoErrors:                     68492
    etherStatsRXNoErrors:                     5016
    ifInMulticastPkts:                        5016
    ifOutBroadcastPkts:                       5790
    ifOutMulticastPkts:                       62702
    ifOutBroadcastPkts:                       5790
    bcmReceivedUndersizePkts:                 0
    bcmTransmittedUndersizePkts:              5790
    bcmQmxDot1dBasePortDelayExceededDiscards: 0
    bcmQmxDot1dBasePortMtuExceededDiscards:   0
    bcmQmxDot1dTpPortInFrames:                5016
    bcmQmxDot1dTpPortOutFrames:               68492
    bcmQmxEtherStatsPkts64Octets:             5790
    bcmQmxEtherStatsPkts128to255Octets:       24
    bcmQmxEtherStatsPkts256to511Octets:       0
    bcmQmxEtherStatsPkts512to1023Octets:      0
    bcmQmxEtherStatsPkts1024to1518Octets:     0
    bcmQmxEtherRxOversizePkts:                0
    bcmQmxEtherTxOversizePkts:                0
    bcmQmxEtherStatsJabbers:                  0
    bcmQmxEtherStatsCRCAlignErrors:           0
    bcmQmxDot3StatsFCSErrors:                 0
    bcmQmxDot3StatsSingleCollisionFrames:     0
    bcmQmxDot3StatsMultipleCollisionFrames:   0
    bcmQmxDot3StatsSQETTestErrors:            0
    bcmQmxDot3StatsDeferredTransmissions:     0
    bcmQmxDot3StatsLateCollisions:            0
    bcmQmxDot3StatsExcessiveCollisions:       0
    bcmQmxDot3StatsInternalMacTransmitErrors: 0
    bcmQmxDot3StatsCarrierSenseErrors:        0
    bcmQmxDot3StatsFrameTooLongs:             0
    bcmQmxDot3StatsInternalMacReceiveErrors:  0
    bcmQmxDot3StatsSymbolErrors:              0
```

```
   bcmQmxDot3ControlInUnknownOpcodes:        0
   bcmQmxDot3InPauseFrames:                  0
   bcmQmxDot3OutPauseFrames:                 0
   bcmQmxIfHCInOctets:                       511632
   bcmQmxIfHCInUcastPkts:                    0
   bcmQmxIfHCInMulticastPkts:                5016
   bcmQmxIfHCInBroadcastPkts:                0
   bcmQmxIfHCOutOctets:                      6236484
   bcmQmxIfHCOutUcastPkts:                   0
   bcmQmxIfHCOutMulticastPkts:               62702
   bcmQmxIfHCOutBroadcastPckts:              5790
   bcmQmxIeee8021PfcRequests:                0
   bcmQmxIeee8021PfcIndications:             0
   bcmQmxBcmEtherStatsPkts1519to1522Octets:  0
   bcmQmxBcmEtherStatsPkts1522to2047Octets:  0
   bcmQmxBcmReceivedPkts64Octets:            0
   bcmQmxBcmReceivedPkts65to127Octets:       5016
   bcmQmxBcmReceivedPkts128to255Octets:      0
   bcmQmxBcmReceivedPkts256to511Octets:      0
   bcmQmxBcmReceivedPkts512to1023Octets:     0
   bcmQmxBcmReceivedPkts1024to1518Octets:    0
   bcmQmxBcmReceivedPkts1519to2047Octets:    0
   bcmQmxBcmTransmittedPkts64Octets:         5790
   bcmQmxBcmTransmittedPkts65to127Octets:    62678
   bcmQmxBcmTransmittedPkts128to255Octets:   24
   bcmQmxBcmTransmittedPkts256to511Octets:   0
   bcmQmxBcmTransmittedPkts512to1023Octets:  0
   bcmQmxBcmTransmittedPkts1024to1518Octets: 0
   bcmQmxBcmTransmittedPkts1519to2047Octets: 0
   bcmQmxBcmTransmittedPkts2048to4095Octets: 0
   bcmQmxBcmTransmittedPkts4095to9216Octets: 0

Logical Interface:  ifl-0/0/10/100, Physical Interface:  ifp-0/0/10
   Counter               Direction   Unit       Rx         Rx Diff    Rx Rate    Tx         Tx Diff   Tx Rate
   IPv4                  -           Packets    -          -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   IPv6                  -           Packets    -          -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   MPLS                  -           Packets    -          -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   Punt                  -           Packets    -          -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   Miss                  RX          Packets    -          -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   Drops                 -           Packets    4995       -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   Error                 RX          Packets    -          -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   Error                 TX          Packets    47         -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   No Buff               RX          Packets    -          -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   Traffic Statistics    -           Packets    4995       -          -          68492      -          -
                                     Bytes      489510     -          -          5869876    -          -
   Unicast Statistics    -           Packets    -          -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   Broadcast Statistics  -           Packets    -          -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   Multicast Statistics  -           Packets    -          -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   Packet Statistics:
     Ingress Forwarded Packets: 1810
     Ingress Forwarded Bytes:   184620
     Ingress Drop Packets:      1
     Ingress Drop Bytes:        102
     Egress Forwarded Packets:  0
     Egress Forwarded Bytes:    0
     Egress Drop Packets:       0
     Egress Drop Bytes:         0
Logical Interface:  ifl-0/0/10/200, Physical Interface:  ifp-0/0/10
   Counter               Direction   Unit       Rx         Rx Diff    Rx Rate    Tx         Tx Diff   Tx Rate
   IPv4                  -           Packets    -          -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   IPv6                  -           Packets    -          -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   MPLS                  -           Packets    -          -          -          -          -          -
```

```
                                     Bytes      -          -          -          -          -          -
   Punt                  -           Packets    -          -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   Miss                  RX          Packets    -          -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   Drops                 -           Packets    -          -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   Error                 RX          Packets    -          -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   Error                 TX          Packets    -          -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   No Buff               RX          Packets    -          -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   Traffic Statistics    -           Packets    -          -          -          6811       -          -
                                     Bytes      -          -          -          573170     -          -
   Unicast Statistics    -           Packets    -          -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   Broadcast Statistics  -           Packets    -          -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   Multicast Statistics  -           Packets    -          -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   Packet Statistics:
     Ingress Forwarded Packets: 0
     Ingress Forwarded Bytes:   0
     Ingress Drop Packets:      0
     Ingress Drop Bytes:        0
     Egress Forwarded Packets:  0
     Egress Forwarded Bytes:    0
     Egress Drop Packets:       0
     Egress Drop Bytes:         0
Logical Interface:  ifl-0/0/10/300, Physical Interface:  ifp-0/0/10
   Counter               Direction   Unit       Rx         Rx Diff    Rx Rate    Tx         Tx Diff   Tx Rate
   IPv4                  -           Packets    -          -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   IPv6                  -           Packets    -          -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   MPLS                  -           Packets    -          -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   Punt                  -           Packets    -          -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   Miss                  RX          Packets    -          -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   Drops                 -           Packets    -          -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   Error                 RX          Packets    -          -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   Error                 TX          Packets    -          -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   No Buff               RX          Packets    -          -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   Traffic Statistics    -           Packets    -          -          -          5902       -          -
                                     Bytes      -          -          -          531180     -          -
   Unicast Statistics    -           Packets    -          -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   Broadcast Statistics  -           Packets    -          -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   Multicast Statistics  -           Packets    -          -          -          -          -          -
                                     Bytes      -          -          -          -          -          -
   Packet Statistics:
     Ingress Forwarded Packets: 0
     Ingress Forwarded Bytes:   0
     Ingress Drop Packets:      0
     Ingress Drop Bytes:        0
     Egress Forwarded Packets:  0
     Egress Forwarded Bytes:    0
     Egress Drop Packets:       0
     Egress Drop Bytes:         0
```

Show Port Map Command

The show port map command displays information about a physical interface's mapping with ONL port IDs and optic port IDs.

Syntax:

show port map

Example: Port mapping

```
supervisor@rtbrick>LEAF01: op> show port map
   IFP-Name            Interface-Type      Port-ID   Onlp-Port-ID   Optic-Port-ID
   cpu-0/0/0           cpu                 0         -              -
   ifp-0/0/0           ethernet            1         32             32
   ifp-0/0/1           ethernet            2         33             33
   ifp-0/0/2           ethernet            3         34             34
   ifp-0/0/3           ethernet            4         35             35
   ifp-0/1/1           ethernet            5         1              1
   ifp-0/1/2           ethernet            9         2              2
   ifp-0/1/3           ethernet            13        3              3
   ifp-0/1/4           ethernet            17        4              4
   ifp-0/1/5           ethernet            21        5              5
   ifp-0/1/6           ethernet            25        6              6
   ifp-0/1/7           ethernet            29        7              7
   ifp-0/1/8           ethernet            33        8              8
   ifp-0/1/9           ethernet            37        9              9
   ifp-0/1/10          ethernet            41        10             10
   ifp-0/1/11          ethernet            45        11             11
   ifp-0/1/12          ethernet            49        12             12
  <...>
```

Interface Clear Commands

Clear commands allow to reset operational states.

Interface Statistics

This command clears interface counters.

Syntax:

clear interface statistics <option>

Option	Description
-	Without any additional option, the command clears the counters for all interfaces.
<interface-name>	Clears the counters for the given interface.

5.2. ARP/ND

5.2.1. ARP/ND Overview

RBFS allows you to set the timer information for ARP and ND routes. The timer information specifies how frequently a device sends messages to its neighbor. RBFS provides a logical interface with which you can configure timers for neighbor routers. These timers are essential for protocols such as ARP and ND.

RBFS supports timers for the following attributes:

Gratuitous ARP Interval

The Gratuitous ARP is sent as a broadcast by a node to communicate its IP address to MAC address mapping on the network. The GARP timer enables you to specify the interval time based on which GARP can be communicated.

Neighbor Probe Interval

RBFS allows you to configure the neighbor probe interval for the specified interface. This attribute is used to ensure that the neighbor is available or not.

Router Advertisement Interval

RBFS allows you to configure router advertisement interval. Router advertisement includes route information to show the network hosts that the router is operational. The router sends these messages periodically within a time range specified with minimum and maximum values. The Router Advertisement Interval timer applies only to IPv6.

Neighbor Scan Interval

It specifies the time interval for neighbor router scanning. It scans the ARP table of a neighbor router to determine which IP addresses are active.

ARP Throttle Interval

ARP throttling is a method of rate limiting of ARP packets and it safeguards the router by limiting too many ARP requests triggered by incoming traffic. RBFS allows you to configure the time interval for ARP throttling.

Supported Platforms

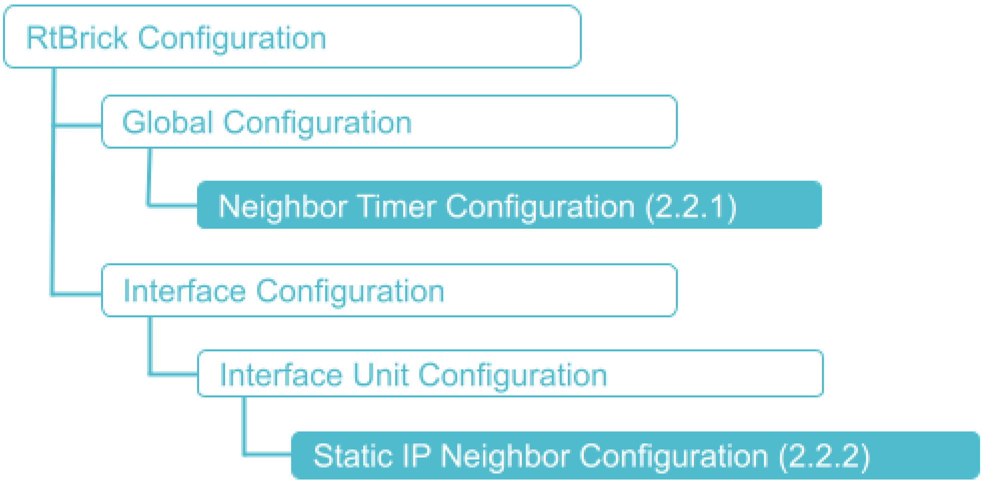
Not all features are necessarily supported on each hardware platform. Refer to the

Platform Guide for the features and the sub-features that are or are not supported by each platform.

5.2.2. ARP/ND Configuration

Configuration Hierarchy

The diagram illustrates the Neighbor Timer configuration hierarchy.


RtBrick Configuration
Global Configuration
Neighbor Timer Configuration (2.2.1)
Interface Configuration
Interface Unit Configuration
Static IP Neighbor Configuration (2.2.2)

Configuration Syntax and Commands

Neighbor Timer Configuration

The following sections describe the interface configuration syntax and commands.

Syntax:

set global neighbor <attribute> <value>

Attribute	**Description**
garp-interval	Gratuitous ARP interval. The value is in seconds. Range: 1 to 1000 seconds Default: 60
probe-interval	Neighbor probe interval. The value is in seconds. Range: 1 to 1000 seconds Default: 10

Attribute	Description
ra-interval	Router advertisement interval. The value is in seconds. Range: 1 to 1000 seconds Default: 10
scan interval	Neighbor scan interval. The value is in seconds. Range: 1 to 1000 seconds Default: 60
throttle-interval	ARP throttle interval. The value is in seconds. Range: 1 to 1000 seconds Default: 10

Example: Neighbor timer Configuration

```
{
    "rtbrick-config:neighbor": {
      "garp-interval": 10,
      "probe-interval": 120,
      "scan-interval": 120,
      "throttle-interval": 120,
      "ra-interval": 120
    }
  }
```

Static IP Neighbor Configuration

This section describes configuration options at static IP neighbors.

Syntax:

set interface <interface-name> **unit** <unit-id> **neighbor** <attribute> <value>

Attribute	Description
<interface-name>	Name of the interface. Examples: ifp-0/0/1.
<unit-id>	Create a logical interface (also referred to as a sub-interface) under the physical interface.
IPv4/IPv6 <ip-address>	Neighbor IPv4 or IPv6 address.
MAC <mac-address>	Neighbor MAC address.

Example: Static IP Neighbor Configuration

```
supervisor@rtbrick>LEAF01: cfg> show config
{
  "data": {
    "rtbrick-config:interface": [
      {
        "name": "ifp-0/1/5",
        "unit": [
          {
            "unit-id": 1,
            "neighbor": {
              "ipv4": [
                {
                  "address4": "198.51.100.10",
                  "mac": "11:11:11:11:11:11"
                }
              ]
            }
          }
        ]
      }
    ]
  }
}
supervisor@rtbrick>LEAF01: cfg>
```

5.2.3. ARP/ND Operational Commands

Show Commands

Neighbor Timer Show Commands

Syntax:

show neighbor <option>

Option	Description
-	Without any option, the commands display the information for all neighbors.
instance <instance_name>	Displays summary of the specified neighbor instance
<afi>	Displays neighbor summary for the specified address family

Example 1: Summary of neighbor

```
supervisor@rtbrick>LEAF01: op> show neighbor
Instance                MAC Address          Interface             IP Address                Dynamic  Entry
Time
default                 e4:ed:7a:8e:d5:9d    ifl-0/0/5/1           2001:db8:0:30::
```

```
true     Thu Feb 24 02:23:19
```

Example 2: Summary of neighbor instance

```
supervisor@rtbrick>LEAF01: op> show neighbor instance default
Instance                MAC Address          Interface             IP Address
Dynamic  Entry Time
default                 7a:01:bf:60:03:02    ifp-0/2/3/10        2001:db8:0:111::
true     Thu Feb 24 04:57:22
default                 7a:01:bf:60:03:02    ifp-0/2/3/20        2001:db8:0:19::
true     Thu Feb 24 04:57:22
```

Example 3: Summary of the neighbor for the specified address family

```
supervisor@rtbrick>LEAF01: op> show neighbor ipv4
  <cr>
  instance              Instance name
```

```
supervisor@rtbrick>LEAF01: op> show neighbor ipv6
Instance                MAC Address          Interface             IP Address                Dynamic  Entry
Time
default                 7a:01:bf:60:03:02    ifp-0/2/3/10        2001:db8:0:111::    true     Thu Feb 24
04:57:22
default                 7a:01:bf:60:03:02    ifp-0/2/3/20        2001:db8:0:19::     true     Thu Feb 24
04:57:22
```

Example 4: Summary of the static IP neighbor

```
supervisor@rtbrick>LEAF01: op> show neighbor
Instance                MAC Address          Interface             IP Address
Dynamic  Entry Time
default                 11:11:11:11:11:11    ifl-0/1/5/1           198.51.100.10
true     Fri Feb 25 10:13:04
```

Neighbor Address Resolution

Syntax:

show address resolution <request | response>

Option	Description
Request	Displays the summary of the address resolution request
Response	Displays the summary of address resolution response

Example 1: Summary of the neighbor address resolution request

```
supervisor@rtbrick>LEAF01: op> show address-resolution request
TableName: global.static.1.address.resolution.request
Next Hop                    AFI         SAFI        Instance
2001:db8:0:30::            ipv6        labeled-un  default
```

Example 2: Summary of the neighbor address resolution response

```
supervisor@rtbrick>LEAF01: op> show address-resolution response
TableName: global.static.1.address.resolution.response
IP Address                  Covering Prefix            MAC Address         Interface
2001:db8:0:30::              2001:db8:0:30::/32        e4:ed:7a:8e:d5:9d   ifl-0/0/5/1
```

5.3. ACLs

5.3.1. ACL Overview

ACL Use Cases

In RBFS, Access Control Lists (ACL) serve multiple purposes:

- Provide security by traffic filtering. This applies to both host and transit traffic. ACLs for traffic filtering are user-defined by configuration.
- Redirecting control traffic to the CPU. Such protocol ACLs also referred to as trap rules, are automatically created by the respective protocol, and do not need to be configured.
- Classifying traffic for differentiated QoS treatment. This is a special form of ACL referred to as a multi-field (MF) classifier. For more information about MF classifiers, please refer to the HQoS Configuration Guide.

ACL Components and Processing

User-defined ACLs consist of rules and ordinals. In case of multiple matching ACL rules, you can use priorities to define the result of the ACL.

- Rules - A rule is a named ACL entry that typically contains one or multiple match criteria and an action.
- Ordinals - An ordinal is solely a numbered configuration object. A rule can consist of multiple ordinals. Ordinals help to structure the configuration. In

RBFS, it makes no difference if you configure one rule with multiple ordinals or multiple rules with one ordinal each. Please note ordinals do not define the order of processing.

- Scope - ACLs generally apply globally. In particular, they are not applied to interfaces. You can, however configure an interface as a match criteria.
- Priorities - ACL entry priorities are used to define the processing of multiple matching ACL rules. In RBFS, by default, all ACL entries have the same priority, and there is no specific order. For example, if one ACL rule shall permit ICMP traffic from a specific prefix, and another rule shall deny any other ICMP traffic, it will by default result in a conflict as an ICMP packet matches both rules. To ensure that the more specific rule matches first, you can set its priority to higher. When the ACL priority value is set to a lower number, priority is higher.

Prefix Lists

A prefix list is a named list of prefixes. Instead of listing multiple individual prefixes in a match rule of the ACL itself, you can reference a list that contains the prefixes, and thereby apply a common action to all matching prefixes. This helps to maintain lists and reuse them in multiple ACL rules.

Prefix lists can be used in ACL for permitting/denying traffic and in Multifield Classifier (MFC) for classifying traffic. This guide describes how to configure prefix lists and apply them in user-defined ACLs as firewall filters and apply prefix lists in MFC for traffic classification. For more information about applying prefix lists to MF classifiers, please refer to the *HQoS configuration Guide*.

When a prefix list is configured and referenced in an ACL, it is internally first added to an intermediate ACL configuration table. For each prefix, one separate rule is added to the final ACL configuration table. This is different from a prefix match in the ACL rule itself that is directly added to the ACL configuration table. A dedicated range of ordinals (200001-4294967295) is reserved to expand ACL rules when using prefix lists. If configured, the priority will be copied from the prefix list ACL configuration to all the expanded ACL rules.

When using prefix lists, the following restrictions apply:

- You cannot configure the same prefix-list name to match the source prefix list and destination prefix list.
- You cannot configure both the source prefix and source prefix list on the same

ACL configuration.

- You cannot configure both the destination prefix and destination prefix list on the same ACL configuration.

Supported Platforms

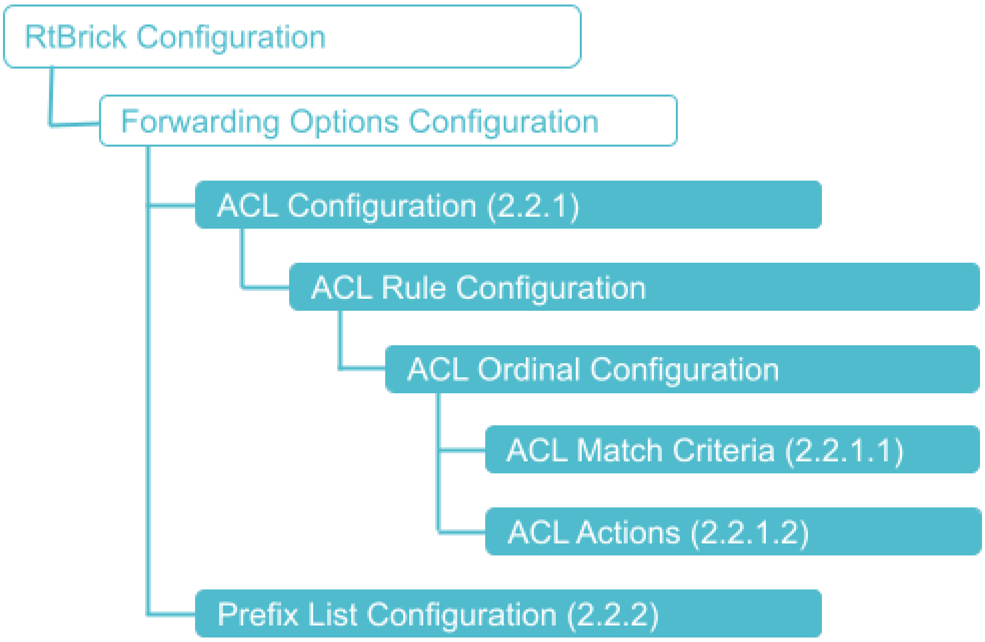
Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

5.3.2. ACL Configuration

For configuring an access control list, you define filter criteria (with match conditions) for the packets and an action for the device to take if the packets match the filtering criteria.

Configuration Hierarchy

The diagram illustrates the ACL configuration hierarchy.


RtBrick Configuration
Forwarding Options Configuration
ACL Configuration (2.2.1)
ACL Rule Configuration
ACL Ordinal Configuration
ACL Match Criteria (2.2.1.1)
ACL Actions (2.2.1.2)
Prefix List Configuration (2.2.2)

Configuration Syntax and Commands

The following sections describe the ACL configuration syntax and commands.

Configuring ACLs

Syntax:

set forwarding-options acl [**l2** | **l3v4** | **l3v6**] **rule** <name> **ordinal** <number> <option> <attribute> <value>

Options	Description
<name>	Name of the ACL rule.
<number>	Specifies the ordinal number.
match <…>	Match configuration hierarchy. Please refer to section 2.2.1.1 for the ACL match criteria configuration.
action <…>	Action configuration hierarchy. Please refer to section 2.2.1.2 for the ACL actions configuration.
priority <priority>	Specifies the ACL priority value. The default entry priority for user-defined ACLs changes to 500. The configurable ACL entry priority range becomes 100 - 20000. A lower number indicates higher priority.

Configuring ACL Match Criteria

set forwarding-options acl [**l2** | **l3v4** | **l3v6**] **rule** <rulename> **ordinal** <ordinal_value> **match** <attribute> <value>

Attribute	Description
destination-mac <destination-mac>	ACL L2 destination mac match.
destination-ipv4-prefix <destination-ipv4-prefix>	ACL L3 IPv4 destination prefix match.
destination-ipv4-prefix-list <destination-ipv4-prefix-list>	ACL destination IPv4 prefix-list name. You can apply a prefix list that is previously configured. Refer to section Configuring Prefix Lists.
destination-l4-port <destination-l4-port>	ACL L4 destination port match.

Attribute	Description
destination-ipv4-local true	Indicates whether match support is enabled for all traffic destined for the routers' IP addresses. **Note**: To disable the configuration, use the delete form of the command.
destination-ipv6-prefix <destination-ipv6-prefix>	ACL L3 IPv6 destination prefix match.
destination-ipv6-prefix-list <destination-ipv6-prefix-list>	ACL destination IPv6 prefix-list name. You can apply a prefix list that is previously configured. Refer to section Configuring Prefix Lists.
destination-ipv6-local true	Indicates whether match support is enabled for all traffic destined for the routers' IP addresses. To disable the configuration, use the delete form of the command.
direction ingress	ACL L2/L3 direction match. Currently, only the ingress direction is supported.
ethertype <ethertype>	ACL L2 EtherType match.
inner-tag-protocol-id <inner-tag-protocol-id>	ACL L2 inner TPID match.
inner-vlan <inner-vlan>	ACL L2 inner-VLAN match.
inner-vlan-cfi <inner-vlan-cfi>	ACL L2 inner-VLAN CFI match.
inner-vlan-priority <inner-vlan-priority>	ACL L2 inner-VLAN priority match.
interface <interface>	Interface match.
ip-options true	Match if the IPv4 packet has options. Supported value: true.
ip-protocol <protocol>	ACL IP protocol value match such as TCP, UDP, ICMP.
ipv4-dscp <ipv4-dscp>	IPv4 DSCP value.
ipv4-tos <ipv4-tos>	IPv4 ToS value.
ipv6-tc <ipv6-tc>	Codepoint class value.
logical-interface <logical-interface>	Logical interface match.

Attribute	Description
source-ipv4-prefix <source-ipv4-prefix>	ACL L3 IPv4 source prefix match.
source-ipv4-prefix-list <source-ipv4-prefix-list>	ACL source IPv4 prefix-list name. You can apply a prefix list that is previously configured. Refer to section Configuring Prefix Lists.
source-ipv6-prefix <source-ipv6-prefix>	Configure ACL L3 IPv6 source prefix match.
source-ipv6-prefix-list <source-ipv6-prefix-list>	ACL source IPv6 prefix-list name. You can apply a prefix list that is previously configured. Refer to section Configuring Prefix Lists.
source-l4-port <source-l4-port>	ACL L4 source port match.
outer-tag-protocol-id <outer-tag-protocol-id>	ACL L2 outer TPID match.
outer-vlan <outer-vlan>	ACL L2 outer-VLAN match.
outer-vlan-cfi <outer-vlan-cfi>	ACL L2 outer VLAN CFI match.
outer-vlan-priority <outer-vlan-priority>	ACL L2 outer VLAN priority match.
source-mac <source-mac>	ACL L2 source MAC match.
traffic-class <class>	Forward class value. Supported values: class-0 to class-7, class-all.
ttl <ttl>	IPv4 time-to-live value.
match-mpls-traffic true	Match single MPLS label termination. To disable, use the delete form of the command.

Example 1: Layer 2 Match Configuration

```
{
    "rtbrick-config:acl": {
       "l2": {
          "rule": [
            {
              "rule-name": "a10nsp-drop-lag-2",
              "ordinal": [
                {
                  "ordinal-value": 1,
                  "match": {
                    "direction": "ingress",
                    "interface": "lag-2",
                    "outer-vlan-priority": 1
                  },
                  "action": {
                    "drop": "true",
                    "statistics": "true"
                  }
                },
```

Example 2: Layer 3 IPv4 Match Configuration

```
{
    "rtbrick-config:acl": {
      "l3v4": {
        "rule": [
          {
            "rule-name": "rtb_firewall_two",
            "ordinal": [
              {
                "ordinal-value": 1000,
                "match": {
                  "direction": "ingress",
                  "source-ipv4-prefix": "198.51.100.50/24",
                  "source-l4-port": 8080
                },
                "action": {
                  "drop": "true"
                }
              }
            ]
          },
          {
            "rule-name": "rule2",
            "ordinal": [
              {
                "ordinal-value": 5,
                "match": {
                  "direction": "ingress",
                  "interface": "ifp-0/0/1"
                }
              }
            ]
          }
        ]
      }
    }
  }
```

Example 3: Layer 3 IPv6 Match Configuration

```
{
    "rtbrick-config:l3v6": {
      "rule": [
        {
          "rule-name": "rtb_firewall_two",
          "ordinal": [
            {
              "ordinal-value": 1000,
              "match": {
                "direction": "ingress",
                "source-ipv6-prefix": "2001:db8:0:11::/32",
                "source-l4-port": 8080
              },
              "action": {
                "permit": "true"
              }
            }
          ]
        }
      ]
    }
  }
```

Example 4: Match support for all traffic destined for any of the router’s IP addresses

```
{
    "rtbrick-config:acl": {
      "l3v4": {
        "rule": [
          {
            "rule-name": "rule4",
            "ordinal": [
              {
                "ordinal-value": 4,
                "match": {
                  "direction": "ingress",
                  "destination-ipv4-local": "true"
                },
                "action": {
                  "drop": "true"
                }
              }
            ]
          }
        ]
      },
      "l3v6": {
        "rule": [
          {
            "rule-name": "rule2",
            "ordinal": [
              {
                "ordinal-value": 2,
                "match": {
                  "direction": "ingress",
                  "destination-ipv6-local": "true"
                },
                "action": {
                  "drop": "true"
                }
              }
            ]
          }
        ]
      }
    }
  }
```

Configuring ACL Actions

Syntax:

set forwarding-options acl [**l3v4** | **l3v6**] **rule** <rulename> **ordinal** <ordinal_value> **action** <attribute> <value>

Attribute	Description
drop true	If the ACL rule specifies drop true, the system will discard any packets that match that rule. Use the delete form of the command for the system to ignore the rule.
permit true	If the ACL rule specifies permit true, the system forwards traffic matching that rule. Use the delete form of the command for the system to ignore the rule.
action statistics true	Configure action, enable statistics. Use the delete form of the command to disable the configuration. ℹ A limited number of counter resources are available in a common pool for user-defined ACLs, protocol ACLs, L3, and L2X logical interfaces.
forward-class <class>	Specifies forward class value (class-0 to class-7, class-all)
mirror <mirror>	Specifies ACL action mirror name. ℹ Currently, ACLs with mirror actions are not supported.

Attribute	Description
capture true	You can enable the capture action when using the RBFS built-in capture feature with an ACL to more granularly specify the traffic to be captured. To disable, use the delete form of the command. For more information and an example, refer to the *RBFS NOC Troubleshooting Guide*.
policer-name <policer-name>	Specifies policer profile name.
redirect-to-cpu true	Configure action, redirect packets to CPU. To disable, use the delete form of the command.

Example

```
{
            "rule-name": "rtb_firewall_two",
            "ordinal": [
              {
                "ordinal-value": 1000,
                "match": {
                  "direction": "ingress",
                  "source-ipv4-prefix": "198.51.100.50/24",
                  "source-l4-port": 8080
                },
                "action": {
                  "drop": "true"
                }
              }
            ]
          }
```

Configuring Prefix Lists

Configuring IPv4/IPv6 Prefix List for ACL and Multifield Classifier

Syntax:

set forwarding-options prefix-list <prefix-list-name> <attribute> <value>

Attribute	Description
<prefix-list-name>	Name of the prefix list, which will be later used to attach with ACL configuration.
ipv4-prefix <ipv4_prefix>	Specifies the IPv4 prefix address.

Attribute	Description
ipv6-prefix <ipv6_prefix>	Specifies the IPv6 prefix address.

Example 1: Prefix List Configuration

```
supervisor@rtbrick>LEAF01: op> show config forwarding-options prefix-list
{
    "rtbrick-config:prefix-list": [
      {
        "prefix-list-name": "ipv4-list",
        "ipv4-prefix": [
          {
            "ipv4-prefix": "198.51.100.50/24"
          },
          {
            "ipv4-prefix": "198.51.101.60/24"
          },
          {
            "ipv4-prefix": "198.51.102.70/24"
          }
        ]
      }
    ]
  }
```

Example 2: Using Prefix-list in Multifield-Classifier

```
supervisor@rtbrick>LEAF01: op> show config forwarding-options prefix-list pta-iptv-multicast
{
  "rtbrick-config:prefix-list": [
    {
      "prefix-list-name": "ipv4-list",
      "ipv4-prefix": [
          {
            "ipv4-prefix": "198.51.100.50/24"
          },
          {
            "ipv4-prefix": "198.51.101.60/24"
          },
          {
            "ipv4-prefix": "198.51.102.70/24"
          }
      ]
    }
  ]
}
```

Example 3: Viewing Multifield-Classifier Details

```
supervisor@rtbrick>LEAF01: op> show config forwarding-options class-of-service
```

```
multifield-classifier acl l3v4 rule pta-triple-play-8queues ordinal 6000
{
  "rtbrick-config:ordinal": [
    {
      "ordinal-value": 6000,
      "match": {
        "destination-ipv4-prefix-list": "ipv4-list"
      },
      "action": {
        "forward-class": "class-1",
        "remark-codepoint": 248
      }
    }
  ]
}
```

5.3.3. ACL Operational Commands

ACL Show Commands

Syntax:

show acl <option>

Option	Description
-	Without any option, this command displays brief information about the access control list (ACL).
detail	Displays detailed information about the access-control list (ACL).
type <acl_type>	Displays detailed information for the specified ACL type.
rule <acl-rule-name>	Displays detailed information for the specified ACL rule name.
ordinal <ordinal>	Displays detailed information for the specified Ordinal
type <acl_type> rule <rule_name>	Displays detailed information for the specified ACL Type and Rule name
type <acl_type> ordinal <ordinal>	Displays detailed information for the specified ACL Type and Ordinal
rule <rule_name> type <acl_type>	Displays detailed information for the specified ACL Type and Rule name

Option	Description
rule <rule_name> ordinal <ordinal>	Displays detailed information for the specified Rule name and Ordinal
ordinal <ordinal> type <acl_type>	Displays detailed information for the specified ACL Type and Ordinal
ordinal <ordinal> rule <rule_name>	Displays detailed information for the specified Rule name and Ordinal
type <acl_type> rule <rule_name> ordinal <ordinal>	Displays detailed information for the specified Ordinal, ACL Type and Rule name
type <acl_type> ordinal <ordinal> rule <rule_name>	Displays detailed information for the specified Ordinal, ACL Type and Rule name
rule <rule_name> ordinal <ordinal> type <acl_type>	Displays detailed information for the specified Ordinal, ACL Type and Rule name
rule <rule_name> type <acl_type> ordinal <ordinal>	Displays detailed information for the specified Ordinal, ACL Type and Rule name
ordinal <ordinal> rule <rule_name> type <acl_type>	Displays detailed information for the specified Ordinal, ACL Type and Rule name
ordinal <ordinal> type <acl_type> rule <rule_name>	Displays detailed information for the specified Ordinal, ACL Type and Rule name

Example 1: Show information about ACLs

```
supervisor@rtbrick>LEAF01: op> show acl
ACL                         Ordinal    Type         Attach Point
rule4                       4          l3v4         -
                            8          l3v4         -
lldp.ifp-0/0/0.trap.rule    -          l2           ifp-0/0/0
lldp.ifp-0/1/0.trap.rule    -          l2           ifp-0/1/0
lldp.ifp-0/1/1.trap.rule    -          l2           ifp-0/1/1
lldp.ifp-0/1/4.trap.rule    -          l2           ifp-0/1/4
lldp.ifp-0/1/5.trap.rule    -          l2           ifp-0/1/5
lldp.ifp-0/1/6.trap.rule    -          l2           ifp-0/1/6
lldp.ifp-0/1/12.trap.rule   -          l2           ifp-0/1/12
lldp.ifp-0/1/13.trap.rule   -          l2           ifp-0/1/13
lldp.ifp-0/1/22.trap.rule   -          l2           ifp-0/1/22
lldp.ifp-0/1/23.trap.rule   -          l2           ifp-0/1/23
```

Example 2: Show detailed information about ACLs

```
supervisor@rtbrick>LEAF01: op> show acl detail
Rule: rule4
  ACL type: l3v4
  Ordinal: 4
    Match:
      Direction: ingress
      Source IPv4 prefix: 198.51.100.35/24
    Action:
      Drop: True
    Result:
      Trap ID: User Defined
    Statistics:
      Units      Total       Accepted    Dropped
      Packets    4           0           4
      Bytes      424         0           424
  Ordinal: 8
    Match:
      Direction: ingress
      Source IPv4 prefix: 198.51.100.45/24
    Action:
      Drop: True
    Result:
      Trap ID: User Defined
    Statistics:
      Units      Total       Accepted    Dropped
      Packets    9           0           9
      Bytes      990         0           990
Rule: lldp.ifp-0/0/0.trap.rule
  ACL type: l2
  Ordinal: -
    Match:
      Attachment point: ifp-0/0/0
      Direction: ingress
      Destination MAC: 01:80:c2:00:00:0e
    Action:
      Redirect to CPU: True
    Result:
      Trap ID: LLDP
    Statistics:
      Units      Total       Accepted    Dropped
      Packets    105         105         0
      Bytes      12915       12915       0
Rule: lldp.ifp-0/1/0.trap.rule
  ACL type: l2
  Ordinal: -
    Match:
      Attachment point: ifp-0/1/0
      Direction: ingress
      Destination MAC: 01:80:c2:00:00:0e
    Action:
      Redirect to CPU: True
    Result:
      Trap ID: LLDP
    Statistics:
      Units      Total       Accepted    Dropped
      Packets    220         220         0
      Bytes      19140       19140       0
```

Example 3: Show detailed information for a specified ACL Rule

```
supervisor@rtbrick>LEAF01: op> show acl rule4
Rule: rule4
  ACL type: l3v4
  Ordinal: 4
    Match:
      Direction: ingress
      Source IPv4 prefix: 198.51.100.35/24
    Action:
      Drop: True
    Result:
      Trap ID: User Defined
    Statistics:
      Units      Total       Accepted    Dropped
      Packets    4           0           4
      Bytes      424         0           424
  Ordinal: 8
    Match:
      Direction: ingress
      Source IPv4 prefix: 198.51.100.45/24
    Action:
      Drop: True
    Result:
      Trap ID: User Defined
    Statistics:
      Units      Total       Accepted    Dropped
      Packets    9           0           9
      Bytes      990         0           990
```

ACL Statistics Commands

> ℹ ACL statistics are currently not supported for PIM, IGMP, and L2TP protocol traffic.

Syntax:

show acl <option> **statistics**

Option	Description
statistics	Displays ACL statistics information
<acl-name> statistics	Displays ACL statistics information for the specified ACL
type <acl_type> statistics	Displays ACL statistics information for the specified ACL type

Option	Description
rule <rule_name> statistics	Displays ACL statistics information for the specified rule name
ordinal <ordinal> statistics	Displays ACL statistics information for the specified ordinal
type <acl_type> rule <rule_name> statistics	Displays ACL statistics information for the specified ACL type and rule name
type <acl_type> ordinal <ordinal> statistics	Displays ACL statistics information for the specified ACL type and ordinal
rule <rule_name> type <acl_type> statistics	Displays ACL statistics information for the specified ACL type and rule name
rule <rule_name> ordinal <ordinal> statistics	Displays ACL statistics information for the specified rule name and ordinal
ordinal <ordinal> type <acl_type> statistics	Displays ACL statistics information for the specified ACL type and ordinal
ordinal <ordinal> rule <rule_name> statistics	Displays ACL statistics information for the specified rule name and ordinal
<acl_type> rule <rule_name> ordinal <ordinal> statistics	Displays ACL statistics information for the specified ordinal, ACL type and rule name
type <acl_type> ordinal <ordinal> rule <rule_name> statistics	Displays ACL statistics information for the specified ordinal, ACL type and rule name
rule <rule_name> ordinal <ordinal> type <acl_type> statistics	Displays ACL statistics information for the specified ordinal, ACL type and rule name
rule <rule_name> type <acl_type> ordinal <ordinal> statistics	Displays ACL statistics information for the specified ordinal, ACL type and rule name
ordinal <ordinal> rule <rule_name> type <acl_type> statistics	Displays ACL statistics information for the specified ordinal, ACL type and rule name

Option	Description
ordinal <ordinal> type <acl_type> rule <rule_name> statistics	Displays ACL statistics information for the specified ordinal, ACL type and rule name

Example 1: Display ACL statistics information

```
supervisor@rtbrick>LEAF01: op> show acl statistics
ACL                             Units      Total       Accepted    Dropped
rule4                           Packets    4           0           4
                                Bytes      424         0           424
rule4                           Packets    9           0           9
                                Bytes      990         0           990
lldp.ifp-0/0/0.trap.rule        Packets    107         107         0
                                Bytes      13161       13161       0
lldp.ifp-0/1/0.trap.rule        Packets    221         221         0
                                Bytes      19227       19227       0
lldp.ifp-0/1/1.trap.rule        Packets    221         221         0
                                Bytes      19227       19227       0
lldp.ifp-0/1/4.trap.rule        Packets    214         214         0
                                Bytes      31672       31672       0
lldp.ifp-0/1/5.trap.rule        Packets    214         214         0
                                Bytes      31672       31672       0
lldp.ifp-0/1/6.trap.rule        Packets    214         214         0
                                Bytes      31672       31672       0
lldp.ifp-0/1/12.trap.rule       Packets    107         107         0
                                Bytes      13375       13375       0
lldp.ifp-0/1/13.trap.rule       Packets    107         107         0
                                Bytes      13375       13375       0
lldp.ifp-0/1/22.trap.rule       Packets    107         107         0
                                Bytes      13375       13375       0
lldp.ifp-0/1/23.trap.rule       Packets    107         107         0
                                Bytes      13375       13375       0
```

Example 2: Display ACL statistics information for the specified ACL

```
supervisor@rtbrick>LEAF01: op> show acl rule4 statistics
ACL         Units      Total       Accepted    Dropped
rule4       Packets    4           0           4
            Bytes      424         0           424
rule4       Packets    9           0           9
            Bytes      990         0           990
```

ACL Clear Commands

Clear commands allow resetting operational states.

Clear ACL Statistics

Syntax:

clear acl <options>

Option	Description
statistics	Clears all the ACL statistics.
type <acl_type> statistics	Clears all ACL statistics for the specified ACL type
rule <rule_name> statistics	Clears all ACL statistics for the specified rule
ordinal <ordinal> statistics	Clears all ACL statistics for the specified ordinal
type <acl_type> rule <rule_name> statistics	Clears all ACL statistics for the specified ACL type and rule
type <acl_type> ordinal <ordinal> statistics	Clears all ACL statistics for the specified ACL type and ordinal
rule <rule_name> type <acl_type> statistics	Clears all ACL statistics for the specified ACL type and rule
rule <rule_name> ordinal <ordinal> statistics	Clears all ACL statistics for the specified rule and ordinal
ordinal <ordinal> type <acl_type> statistics	Clears all ACL statistics for the specified ACL Type and Ordinal
ordinal <ordinal> rule <rule_name> statistics	Clears all ACL statistics for the specified Rule name and Ordinal
type <acl_type> rule <rule_name> ordinal <ordinal> statistics	Clears all ACL statistics for the specified Ordinal, ACL Type and Rule name
type <acl_type> ordinal <ordinal> rule <rule_name> statistics	Clears all ACL statistics for the specified Ordinal, ACL Type and Rule name
rule <rule_name> ordinal <ordinal> type <acl_type> statistics	Clears all ACL statistics for the specified Ordinal, ACL Type and Rule name

Option	Description
rule <rule_name> type <acl_type> ordinal <ordinal> statistics	Clears all ACL statistics for the specified Ordinal, ACL Type and Rule name
ordinal <ordinal> rule <rule_name> type <acl_type> statistics	Clears all ACL statistics for the specified Ordinal, ACL Type and Rule name
ordinal <ordinal> type <acl_type> rule <rule_name> statistics	Clears all ACL statistics for the specified Ordinal, ACL Type and Rule name

Example: Clearing ACL Statistics of a specified ACL Rule

```
supervisor@rtbrick>LEAF01: op> clear acl rule lldp.ifp-0/0/44.trap.rule statistics
Success : command success
supervisor@rtbrick>LEAF01: op>
```

5.4. Port Mirroring

5.4.1. Port Mirroring Overview

Port Mirroring is a method of monitoring network traffic. When you enable port mirroring, the switch sends a copy of all network packets seen on one port to another port, where the packet can be analyzed.

Inbound Mirroring

Inbound mirroring is defined per In-Port, or per In-Port x VLAN. Configurations for six distinct VLAN tags, for any other VLAN tag, and for packets without VLAN tags are supported. The ingress mirroring can be sampled by specifying a probability that a matching packet will be mirrored.

Outbound Mirroring

Outbound mirroring is defined per Out-Port, or per Out-Port x VLAN tag. Configurations for seven distinct VLAN tags are supported.

Guidelines and Limitations

- Up to 15 mirror profiles can be configured.
- The same mirror resources are used for Lawful Interception (LI) and Port Mirroring.
- You can configure a CPU port as destination physical interface port; but if heavy traffic is mirrored, it may impact system performance.
- If physical interface/logical interface goes down, mirror configuration will not be deleted automatically. You need to delete the mirror configuration explicitly.
- Before creating logical interface mirroring, the source logical interface should exist.
- The logical interface should not be deleted during mirroring.
- If you want to mirror traffic to CPU, enable the control plane security features. For more refer, see the *Control Plane Security Guide*.
- Since this is a debugging tool, the save and reload functionality is not supported.

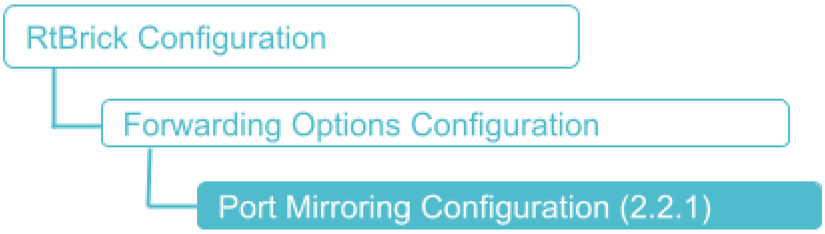
Supported Platforms

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

5.4.2. Port Mirroring Configuration

Configuration Hierarchy

The diagram illustrates the port mirroring configuration hierarchy.


RtBrick Configuration
Forwarding Options Configuration
Port Mirroring Configuration (2.2.1)

Configuration Syntax and Commands

The following sections describe the port mirroring configuration syntax and commands.

Syntax:

set forwarding-options mirror <name> [**source** | **destination**] <attribute> <value>

Attribute	Value
<name>	Name for mirror configuration
acl true	Configure source as ACL. Note: Use delete form of the command to disable the configuration.
direction [egress | ingress]	Configure traffic direction ingress/egress.
interface <interface>	Specifies the physical interface name.
logical-interface <logical-interface>	Configure source logical interface name.

Example 1: Mirroring one physical interface traffic to another physical interface

```
{
    "rtbrick-config:mirror": [
      {
        "name": "MIRROR1",
        "destination": {
          "interface": "ifp-0/0/4"
        },
        "source": {
          "direction": "ingress",
          "interface": "ifp-0/0/2"
        }
      }
    ]
  }
```

Example 2: Mirroring Traffic to CPU

```
{
    "rtbrick-config:mirror": [
      {
        "name": "mirror1",
        "destination": {
          "interface": "cpu-0/0/200"
        },
        "source": {
          "direction": "ingress",
          "interface": "ifp-0/0/52"
        }
      }
    ]
  }
```

5.4.3. Port Mirroring Operational Commands

Capturing Mirror Traffic

After you configure mirroring to CPU by using the commands above, you can use the **capture** command to capture the mirror traffic.

Syntax:

capture mirror file <file_name> [**start** | **stop**]

Attribute	Value
<file_name>	Name of the file where mirror traffic is captured

Example 1: Starting and stopping mirror traffic to a file

```
root@rtbrick: cfg> capture mirror file test.pcap start
root@rtbrick: cfg> capture mirror file test.pcap stop
```

Show Commands

show capture sessions

Syntax:

show capture sessions

Option	Description
-	Without any option, the commands displays the FIB packet Capture sessions.

Example 1: Summary of FIB packet Capture sessions

```
supervisor@rtbrick: op> show capture sessions
Interface                      Direction   File                            Context
ifp-0/0/1                      BOTH        -                               45
```

5.5. OAM Support

5.5.1. OAM Overview

Operations, Administration and Management (OAM) are the processes, activities, tools, and standards involved with performing operational, administrative, and management tasks. RBFS provides the following OAM features that enable you troubleshoot RtBrick software:

- IP ping
- IP traceroute
- IP ping on an MPLS transport
- IP traceroute on an MPLS transport

MPLS ping and traceroute will be supported in the later releases

Supported Platforms

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

5.5.2. OAM Configuration

Guidelines

- Execute the OAM commands in the **operation** mode of the CLI.

```
admin@rtbrick:~$ cli
admin@rtbrick: cfg> switch-mode operation
Activating syntax mode : op [operation]
admin@rtbrick: op>
```

IP Ping

The IP ping utility is used to check the reachability of an IP address.

IP ping in default instance

The **ping** command allows you to ping to a destination to see if a networked device is reachable.

Syntax

```
ping <destination-ip> [source-interface <interface>] count <count> interval
<interval> size <size> source-ip <source-ip> ttl <ttl> tos <tos>
```

Command Parameters

destination-ip	ipv4 destination address
interface	source interface
count	count of the ping packet (default 5)
do-not-fragment	set the do-not-fragment (DF) bit in IP header
interval	interval between the packets (default 1 sec)
size	size of the ping packet (default 60)
source-ip	IPv4 source address
ttl	Time-to-live to be used in the packet
tos	Type of Service (TOS) to be used in the packet

Example

```
admin@rtbrick: op> ping 198.51.100.111
68 bytes from 198.51.100.111: icmp_seq=1 ttl=64 time=11.8707 ms
68 bytes from 198.51.100.111: icmp_seq=2 ttl=64 time=1.9824 ms
68 bytes from 198.51.100.111: icmp_seq=3 ttl=64 time=5.0726 ms
68 bytes from 198.51.100.111: icmp_seq=4 ttl=64 time=5.6529 ms
68 bytes from 198.51.100.111: icmp_seq=5 ttl=64 time=10.6588 ms
Statistics: 5 sent, 5 received, 0% packet loss
```

IP ping in specific instance and afi/safi

This command allows you to ping to a destination in a particular VRF.

Syntax

```
ping <destination-ip> [instance <instance-name> afi <afi> safi<safi>] [source-
interface <interface>] count <count> interval <interval> size <size> source-ip
<source-ip> ttl <ttl> tos <tos>
```

Command Parameters

destination-ip	ipv4 destination address
instance-name	instance on which ping has to be executed
afi	IPv4 Address Family Identifier (AFI)
safi	Subsequent address family identifier (SAFI)
interface	source interface
count	count of the ping packet (default 5)
do-not-fragment	set the do-not-fragment (DF) bit in IP header
interval	interval between the packets (default 1 sec)
size	size of the ping packet (default 60)
source-ip	IPv4 source address
ttl	Time-to-live to be used in the packet
tos	Type of Service (TOS) to be used in the packet

The afi/safi attributes are optional; if not specified, afi would be ipv4 and safi would be unicast.

Example

```
admin@rtbrick: op> ping 198.51.100.80 instance ip2vrf afi ipv4 safi labeled-
unicast
68 bytes from 198.51.100.80: icmp_seq=1 ttl=64 time=18.1306 ms
68 bytes from 198.51.100.80: icmp_seq=2 ttl=64 time=32.1058 ms
68 bytes from 198.51.100.80: icmp_seq=3 ttl=64 time=19.8205 ms
68 bytes from 198.51.100.80: icmp_seq=4 ttl=64 time=20.0144 ms
68 bytes from 198.51.100.80: icmp_seq=5 ttl=64 time=32.0085 ms
Statistics: 5 sent, 5 received, 0% packet loss
```

IPv6 Ping

The IPv6 ping utility is used to check the reachability of an IPv6 address.

IPv6 ping in default instance

The **ping** command allows you to ping to an IPv6 destination to see if a networked device is reachable.

Syntax

```
ping <destination-ipv6> [source-interface <interface>] count <count> interval
<interval> size <size> source-ip <source-ipv6> ttl <ttl> tos <tos>
```

Command Parameters

destination-ipv6	ipv6 destination address
interface	source interface
count	count of the ping packet (default 5)
do-not-fragment	set the do-not-fragment (DF) bit in IP header
interval	interval between the packets (default 1 sec)
size	size of the ping packet (default 60)
source-ipv6	IPv6 source address
ttl	Time-to-live to be used in the packet
tos	Type of Service (TOS) to be used in the packet

Example

```
admin@rtbrick: op> ping 2001:db8:0:42::
68 bytes from 2001:db8:0:42::: icmp_seq=1 ttl=64 time=.0503 ms
68 bytes from 2001:db8:0:42::: icmp_seq=2 ttl=64 time=.0321 ms
68 bytes from 2001:db8:0:42::: icmp_seq=3 ttl=64 time=.0314 ms
68 bytes from 2001:db8:0:42::: icmp_seq=4 ttl=64 time=.0325 ms
68 bytes from 2001:db8:0:42::: icmp_seq=5 ttl=64 time=.0354 ms
Statistics: 5 sent, 5 received, 0% packet loss
```

IPv6 ping in specific instance and afi/safi

This command allows you to ping to an IPv6 destination in a particular VRF.

Syntax

```
ping <destination-ipv6> [instance <instance-name> afi<afi> safi<safi>] [source-
interface <interface>] count <count> interval <interval> size <size> source-ip
<source-ipv6> ttl <ttl> tos <tos>
```

Command Parameters

destination-ipv6	ipv6 destination address
instance-name	instance on which ping has to be executed
afi	IPv4 Address Family Identifier (AFI)
safi	Subsequent address family identifier (SAFI)
interface	source interface
count	count of the ping packet (default 5)
do-not-fragment	set the do-not-fragment (DF) bit in IP header
interval	interval between the packets (default 1 sec)
size	size of the ping packet (default 60)
source-ipv6	IPv6 source address
ttl	Time-to-live to be used in the packet
tos	Type of Service (TOS) to be used in the packet

ℹ The afi/safi attributes are optional; if not specified, afi would be ipv6 and safi would be unicast.

Example

```
admin@rtbrick: op> ping 2001:db8:0:42:: instance abc afi ipv6 safi labeled-unicast
68 bytes from 2001:db8:0:42::: icmp_seq=1 ttl=64 time=.0503 ms
68 bytes from 2001:db8:0:42::: icmp_seq=2 ttl=64 time=.0321 ms
68 bytes from 2001:db8:0:42::: icmp_seq=3 ttl=64 time=.0314 ms
68 bytes from 2001:db8:0:42::: icmp_seq=4 ttl=64 time=.0325 ms
68 bytes from 2001:db8:0:42::: icmp_seq=5 ttl=64 time=.0354 ms
Statistics: 5 sent, 5 received, 0% packet loss
```

IP traceroute

IP traceroute in default instance

This command allows you to traceroute to a particular IP destination.

Syntax

```
traceroute <destination-ip> [source-interface <interface>] repeat <repeat>
interval <interval> size <pktsize> source-ip <source-ip> maxhop <maxhop>
```

destination-ip	ipv4 destination address
interface	source interface
repeat	no of packets for each hop (default 3)
interval	interval between the packets (default 1 sec)
pktsize	size of the traceroute packet (default 60)
source-ip	source IP address
maxhop	max number of hops before the TTL expires (default 30)

Example

```
admin@rtbrick: op> traceroute 198.51.100.80
traceroute to 198.51.100.80, 30 hops max, 60 byte packets
1    198.51.100.90    39.401 ms      19.919 ms      20.074 ms
2    198.51.100.80    55.544 ms      36.765 ms      45.989 ms
```

IP traceroute in specific instance and afi/safi

This command allows you to traceroute to a particular IP destination in a specific VRF.

Syntax

```
traceroute <destination-ip> [instance <instance-name> afi <afi> safi<safi>]
[source-interface <interface>] repeat <repeat> interval <interval> size <pktsize>
source-ip <source-ip> maxhop <maxhop>
```

destination-ip	ipv4 destination address
instance-name	instance on which traceroute has to be executed
afi	IPv4 Address Family Identifier (AFI)
safi	Subsequent address family identifier (SAFI)
interface	source interface

repeat	no of packets for each hop (default 3)
interval	interval between the packets (default 1 sec)
pktsize	size of the traceroute packet (default 60)
source-ip	source IP address
maxhop	max number of hops before the TTL expires (default 30)

> The afi/safi attributes are optional; if not specified, afi would be ipv4 and safi would be unicast.

Example

```
supervisor@S1-STD-28-2901>bm13-tst.fsn.rtbrick.net: cfg> traceroute 198.51.100.80
instance default afi ipv4 safi labeled-unicast source-interface
 ifl-0/0/0/1 source-ip 198.51.100.55
traceroute to 198.51.100.80 30 hops max, 60 byte packets
1    198.51.100.12    4.961 ms   .421 ms   .503 ms
     MPLS Label=20071  Exp=0  TTL=1  S=1
2    198.51.100.80    .995 ms   6.456 ms   .813 ms
```

> For the MPLS transport, it displays the ingress labels in each hop.

IPv6 traceroute

IPv6 traceroute in default instance

This command allows you to traceroute to a particular IP destination.

Syntax

```
traceroute <destination-ipv6> [source-interface <interface>] repeat <repeat>
interval <interval> size <pktsize> source-ip <source-ipv6> maxhop <maxhop>
```

destination-ipv6	ipv6 destination address
interface	source interface
repeat	no of packets for each hop (default 3)
interval	interval between the packets (default 1 sec)
pktsize	size of the traceroute packet (default 60)

source-ip	IPv6 source address
maxhop	max number of hops before the TTL expires (default 30)

Example

```
admin@rtbrick: op> traceroute 2001:db8:0:75::
traceroute to 2001:db8:0:75:: 30 hops max, 60 byte packets
1 2001:db8:0:90:: 21.247 ms 20.232 ms 20.052 ms
2 2001:db8:0:75:: 50.124 ms 59.822 ms 40.032 ms
```

IPv6 traceroute in specific instance and afi/safi

This command allows you to traceroute to a particular IPv6 destination in a specific VRF.

Syntax

```
traceroute <destination-ipv6> [instance <instance-name> afi<afi> safi<safi>]
[source-interface <interface>] repeat <repeat> interval <interval> size <pktsize>
source-ip <source-ipv6> maxhop <maxhop>
```

destination-ipv6	ipv6 destination address
instance-name	instance on which traceroute has to be executed
afi	IPv4 Address Family Identifier (AFI)
safi	Subsequent address family identifier (SAFI)
interface	source interface
repeat	no of packets for each hop (default 3)
interval	interval between the packets (default 1 sec)
pktsize	size of the traceroute packet (default 60)
source-ip	IPv6 source address
maxhop	max number of hops before the TTL expires (default 30)

The afi/safi attributes are optional; if not specified, afi would be ipv6 and safi would be unicast.

Example

```
supervisor@S1-STD-28-2901>bm13-tst.fsn.rtbrick.net: cfg> traceroute
2001:db8:0:75:: instance default afi ipv6 safi labeled-unicast source-interface
 ifl-0/0/0/1 source-ip 2001:db8:0:11::
traceroute to 2001:db8:0:75:: 30 hops max, 60 byte packets
1    2001:db8:0:90::    6.782 ms   .306 ms   6.380 ms
     MPLS Label=20072  Exp=0  TTL=1  S=1
2    2001:db8:0:75::    .434 ms   .657 ms   .844 ms
```

For the MPLS transport, it displays the ingress labels in each hop.

5.6. LAG

5.6.1. LAG Overview

A link aggregation group (LAG) combines multiple physical links into a single logical interface which is referred to as a bundle interface. These physical links are connected between two devices. The device uses LACP protocol to bundle the member links and create high speed connections. Although a bundle can be created based on static configuration, bandwidth can be increased by adding member links to the bundle. This also allows load sharing among the physical links. Thus, a group of ports combined together is called a link aggregation group, or LAG.

The LAG interface combines the bandwidth of the individual member links. The properties like speed and bandwidth of the individual member links should be the same to make it part of that LAG. The traffic which is directed towards the LAG interface is sent on the individual member links. This traffic is not pinned to a specific member link but rather determined by a specific flow. This hash could be calculated based on various fields in the packet.

Supported Platforms

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

LAG Interface Modes

The LAG interface could be formed statically or dynamically. LACP protocol helps to bring up the interface dynamically. The two modes of LAG interface are:

1. **Static LAG**: In this mode, the member links do not initiate or process any of the LACP packets received. The device brings up the LAG interface without LACP negotiation.
2. **Dynamic LAG**: In this mode, the member links process the LACP packets received. Under this mode, there two sub modes:
   a. **active**: LACP packets are generated on each of the member links on the transmit side.
   b. **passive**: LACP packets are generated on the member link in response to the LACP packet received. That means, at least one side of the LAG should be configured as active to bring up the LAG interface.

Layer2 and Layer 3 Interfaces

LAG interfaces can be used as layer 2 and layer 3 interfaces. A regular layer 2 or layer 3 interface can be created on top of the single LAG interface. These interfaces can be divided based on 802.1q VLAN ID's. Multiple layer 3 interfaces can be created and each of them can be associated with different instances.

LACP (Link Aggregation Control Protocol)

LACP is part of an IEEE specification (802.3ad) that allows several physical ports to be grouped to form a single logical interface. LACP allows a switch to negotiate an LAG by sending LACP packets on its member links. It negotiates the various configuration parameters to bring up the individual member links.

Supported Number of LAG Interfaces on Platforms

The following tables provide maximum number of LAG interfaces and members per LAG supported on each hardware platform.

Access-Leaf Platforms

Platform	Max. Number of LAG Interfaces Supported	Number of LAG Members Supported
UfiSpace S9600-72XC	72	10
EdgeCore AGR420	74	10

Consolidated BNG Platforms

Platform	Number of LAG Interfaces Supported	Number of LAG Members Supported
UfiSpace S9500-22XST	22	10
Edgecore CSR320	24	10
UfiSpace S9510-28DC	NA	NA
Edgecore AGR420	24	10

Spine Platforms

Platform	Max. Number of LAG Interfaces Supported	Number of LAG Members Supported
UfiSpace S9600-32X	35	10
EdgeCore AGR400	33	10

L2 Wholesale (L2BSA) Platforms

Platform	Max. Number of LAG Interfaces Supported	Number of LAG Members Supported
UfiSpace S9500-22XST	22	10
Edgecore CSR320	24	10

Guidelines and Limitations

- You cannot configure logical interfaces on a LAG member ports.
- You cannot configure L2X on a LAG member port.

5.6.2. LAG Configuration

Creating LAG Interfaces

Syntax:

```
set link-aggregation interface <name> <attribute> <value>
```

Attribute	Description
<name>	Specifies the name of the LAG interface. The supported LAG interface names: 'lag-1' to 'lag-99'.
<description>	Link aggregation interface description
mode <mode>	Specifies the LAG mode. The default mode is LACP. The possible modes are: • lacp - In this mode, the member links processes LACP packets received. When you create a LAG interface in LACP mode, the LACP PDUs are sent and received through member interfaces. • static: In this mode, the member links do not initiate or process any of the LACP packets received.
<minimum-link-count>	Specify the minimum number of active member links required for the link aggregation interface.
<member-interface>	Specify name of the member interface.
redundancy-session-id	Specify the value for the redundancy group session ID. Range from 1 to 65535 is allowed.
system-id	Specify the MAC address (as system ID) of the device for the link-aggregation interface.

redundancy-session-id, and system-id attributes can only be used when you deploy RBFS in redundancy mode. For information about LAG configuration when deploying RBFS in redundancy mode, see RBFS Redundancy Solution Guide.

Example: LAG Interfaces Configuration

```
supervisor@rtbrick: cfg> show config link-aggregation
{
    "rtbrick-config:link-aggregation": {
      "interface": [
        {
          "interface-name": "lag-3",
          "mode": "lacp",
          "minimum-link-count": 2,
          "member-interface": [
            {
              "member-interface-name": "ifp-0/0/1",
              "lacp-mode": "active"
            },
            {
              "member-interface-name": "ifp-0/0/5",
              "lacp-mode": "active"
            }
            {
              "member-interface-name": "ifp-0/0/5",
              "lacp-mode": "passive"
            }
          ]
        }
      ]
    }
  }
```

Configuring LAG Member Interfaces

You can add member ports to the LAG interface. The command below allows you to bundle multiple physical interfaces with similar properties like speed, MTU, MRU.

Syntax:

```
set link-aggregation interface <name> member-interface <name>
<attribute> <value>
```

Attribute	Description
lacp-mode <mode>	Specifies the LACP mode. The default lacp-mode is active. **active**: LACP packets are generated on each of the member links on the trad, the receive side. **passive**: LACP packets are generated on the member link in response to the LACP packet received at one side of the LAG should be configured as active to bring the LAG interface.

Attribute	Description
lacp-timeout <timeout-value>	Specifies the timeout for the LACP session. A long timeout is 90 seconds, while a short is 3 seconds (default is short). Setting the timeout value will instruct the partner at which interval it should send the updates (30 seconds for long timeout, 1 second for short timeout). ℹ Having mismatching timeouts will not break the operation, even though it is not desirable design-wise. This is because in LACP both Actor and Partner negotiate the transmission rate, so the transmitter sends at the receiver’s expected interval.

Configuring QoS on LAG Interface

RBFS supports QoS at physical interface level for LAG. Users can apply QoS profile at physical interface level through which one common QoS classification can be applied for all traffic on that port, irrespective of the destination logical interface.

The following features are supported:

- Classification (IEEE-802.1)
- Remarking (IEEE-802.1)
- Ingress Policing
- Egress Policing

For information about configuring the above features, refer the *HQoS Configuration Guide*.

ℹ

- You cannot apply QoS class of service on LAG logical interface
- Currently, queuing and scheduling are not supported

Syntax:

```
set interface <physical interface> class-of-service <class-of-service>
```

Attribute	Description
<interface>	Name of the interface
<class-of-service>	Specifies the class of service

Example:

```
supervisor@rtbrick: cfg> set interface lag-11 class-of-service Retail_profile
supervisor@rtbrick: cfg> commit
supervisor@rtbrick: cfg> show config int lag-11
{
    "rtbrick-config:interface": [
      {
        "name": "lag-11",
        "class-of-service": "Retail_profile"
      }
    ]
  }
```

Configuring L2X on LAG Interface

All forms of L2X that are supported on the regular physical interfaces are supported on LAG. The incoming packet is be matched to a specific L2X profile based on the Cross Connect configuration on the specified LAG interface.

The following match conditions are supported on the LAG interface:

- Incoming LAG interface without any VLAN
- Incoming LAG interface with a single VLAN
- Incoming LAG interface with inner and Outer VLAN
- Incoming LAG interface with any single VLAN
- Incoming LAG interface with inner VLAN and any outer VLAN

For information about configuring L2X, see the L2X Configuration Guide.

The following table provides the L2X match action attributes which are supported on LAG interface.

Attribute	Description
nexthop6 <nexthop>	Next-Hop address
match-type <match-type>	Match types with which traffic can be matched.
service-label <service_label>	Service label value. NOTE: Supported MPLS label values are 0 - 1048575. The reserved MPLS label range is 0 - 15. In RBFS, BGP uses the label range 20000 - 100000. It is recommended to assign label values outside of these reserved ranges to avoid conflicts.
ingress-vlan-operation <ingress-vlan-action>	VLAN operation on ingress side outer VLAN
ingress-outer-vlan <vlan-id>	Outer VLAN at ingress side
outgoing_ifp	Outgoing interface
vlan_operation	VLAN operation
outgoing_outer_vlan1	Outgoing outer VLAN

5.6.3. LAG Operational Commands

Show Commands

Viewing LAG Running Configuration

The following command displays the LAG running configuration on the system.

Syntax:

```
show config link-aggregation
```

Example: LAG Running Configuration

```
supervisor@dev1: cfg> show config link-aggregation
{
  "rtbrick-config:link-aggregation": {
    "interface": [
      {
        "interface-name": "lag-4",
        "mode": "lacp",
        "minimum-link-count": 4,
        "member-interface": [
          {
            "member-interface-name": "ifp-0/0/1",
            "lacp-mode": "active",
            "lacp-timeout": "long"
          },
          {
            "member-interface-name": "ifp-0/0/4",
            "lacp-mode": "active",
            "lacp-timeout": "long"
          }
        ]
      }
    ]
  }
}
```

Viewing LAG Information

The following command displays the LAG information.

Syntax:

show lag <options>

Option	Description
<interface-name>	Displays information for a specific LAG interface
detail	Displays detailed LAG information
mode <mode>	Displays information for a LAG mode: static or LACP

Example: Viewing LAG Information

```
supervisor@rtbrick: cfg> show lag detail

  Lag interface name: lag-3
  Status:             Up
  Minimum link count: 2
  Mode:               lacp
    Member interface name: ifp-0/0/1
      Actor system id: 04:f8:f8:e9:bc:83
      Actor key: 107
      Partner system id: 04:f8:f8:e9:bf:83
      Partner key: 43
    Member interface name: ifp-0/0/5
      Actor system id: 04:f8:f8:e9:bc:83
      Actor key: 107
      Partner system id: 04:f8:f8:e9:bf:83
      Partner key: 43
```

Viewing LAG QoS Policer Counters

The following command displays the QoS Policer Counters.

Syntax:

```
show qos policer counter
```

Example 1: QoS policer counter

```
supervisor@rtbrick: cfg> show qos policer counter
Interface                         Level  Units      Total            Received
Dropped
lag-27                            1      Packets    0                0
0
                                         Bytes      0                0
0
lag-27                            2      Packets    0                0
0
                                         Bytes      0                0
0
lag-27                            3      Packets    0                0
0
                                         Bytes      0                0
0
lag-27                            4      Packets    0                0
0
                                         Bytes      0                0
0
lag-28                            1      Packets    0                0
0
                                         Bytes      0                0
0
lag-28                            2      Packets    0                0
0
                                         Bytes      0                0
0
lag-28                            3      Packets    0                0
0
                                         Bytes      0                0
0
lag-28                            4      Packets    203              0
203
                                         Bytes      18270            0
18270
lag-29                            1      Packets    0                0
0
                                         Bytes      0                0
0
lag-29                            2      Packets    0                0
0
                                         Bytes      0                0
0
lag-29                            3      Packets    20591812         18850600
```

```
1741212
                                         Bytes      21291933608      19491520400
1800413208
lag-29                            4      Packets    0                0
0
                                         Bytes      0                0
0
lag-27-egress                     1      Packets    0                0
0
                                         Bytes      0                0
0
lag-27-egress                     2      Packets    0                0
0
                                         Bytes      0                0
0
lag-27-egress                     3      Packets    2011928          2011928
0
                                         Bytes      2116548256       2116548256
0
lag-27-egress                     4      Packets    180377           180377
0
                                         Bytes      15574914         15574914
0
lag-28-egress                     1      Packets    0                0
0
                                         Bytes      0                0
0
lag-28-egress                     2      Packets    0                0
0
                                         Bytes      0                0
0
lag-28-egress                     3      Packets    2019661          2019661
0
                                         Bytes      2124683372       2124683372
0
lag-28-egress                     4      Packets    178226           178226
0
                                         Bytes      15398532         15398532
0
lag-29-egress                     1      Packets    0                0
0
                                         Bytes      0                0
0
lag-29-egress                     2      Packets    0                0
0
                                         Bytes      0                0
0
lag-29-egress                     3      Packets    1999328          1999328
0
                                         Bytes      2103293056       2103293056
0
lag-29-egress                     4      Packets    183300           183300
0
                                         Bytes      15893128         15893128
0
```

Syntax:

show qos policer counter <lag interface>

Example 2: QoS policer counter for a specified LAG

```
supervisor@rtbrick: cfg> show qos policer counter lag-29
Interface                         Level  Units      Total            Received
Dropped
lag-29                            1      Packets    0                0
0
                                         Bytes      0                0
0
lag-29                            2      Packets    0                0
0
                                         Bytes      0                0
0
lag-29                            3      Packets    21241103         19499891
1741212
                                         Bytes      21963300502      20162887294
1800413208
lag-29                            4      Packets    0                0
0
                                         Bytes      0                0
0
```

Example 3

```
supervisor@rtbrick: cfg> show qos policer counter lag-29-egress
Interface                         Level  Units      Total            Received
Dropped
lag-29-egress                     1      Packets    0                0
0
                                         Bytes      0                0
0
lag-29-egress                     2      Packets    0                0
0
                                         Bytes      0                0
0
lag-29-egress                     3      Packets    2071701          2071701
0
                                         Bytes      2179429452       2179429452
0
lag-29-egress                     4      Packets    187300           187300
0
                                         Bytes      16221128         16221128
0
```

5.7. HQoS

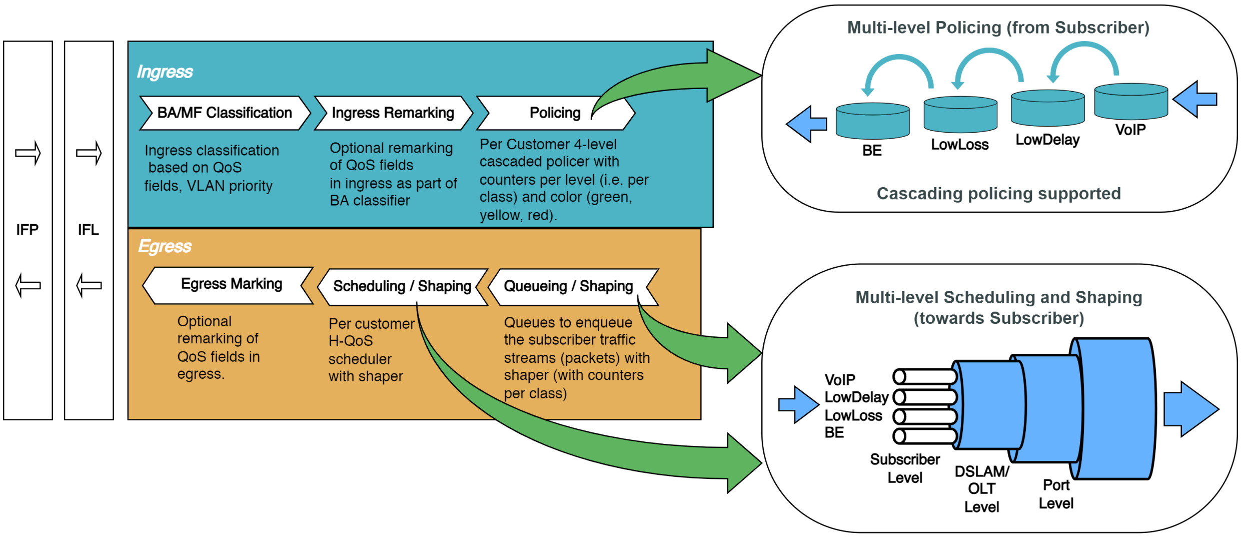
5.7.1. Hierarchical Quality of Service (HQoS) Overview

Hierarchical Quality of Service (HQoS) is a traffic classification and prioritization framework that allows you to provide different Service Level Agreements(SLAs) on bandwidth usage. It allocates network resources to services on a prioritized basis. This is achieved by classifying, policing, shaping, scheduling and remarking the traffic based on service types. The HQoS service can be applied to Ethernet, IPv4, IPv6 or MPLS packets. The framework supports QoS at multiple levels. At only one level it allows you to allocate resources between services, but when configured at multiple levels through HQoS, it allows complex prioritization schemes.

The RtBrick Full Stack (RBFS) uses the following HQoS mechanisms:

- **Classifier:** Classifies each incoming packet as belonging to a specific class, based on packet contents. Supported classifiers are Behavior Aggregate (BA) and Multifield (MF). In the BA classifier, packets are classified according to the CoS field: IEEE 802.1p, IPv4/v6 ToS/TC, or MPLS EXP. In the MF classifier, packets are classified using additional fields in the IP header: source IPv4/IPv6 prefix, destination IPv4/IPv6 prefix, L4 source port, L4 destination port, and/or IP protocol.
- **Policer:** Policer is implemented in the ingress to drop the unwanted traffic. Policer supports the Committed Information Rate (CIR), the Committed Burst Size (CBS), the Peak Information Rate (PIR), and the Peak Burst Size (PBS). Drop behavior is to either mark traffic as green, yellow, or drop.
- **Queuing:** Drop unqualified packets in advance using the Weighted Random Early Detection (WRED) technology in the case of congestion to ensure bandwidth for qualified services. This is performed at the egress.
- **Scheduler:** Manage traffic on a device using different algorithms for queue scheduling. Such algorithms include Fair Queuing (FQ), Weighted Round Robin (WRR), and Strict Priority (SP). Queues can be connected to schedulers and schedulers can also be connected to other schedulers.
- **Shaper:** Shaper is implemented in the egress, traffic shaping involves buffering and delaying traffic to shape the flows. Shapers are implemented either on queues or schedulers.
- **Remarking:** Remarking allows you to rewrite the outgoing packet’s codepoint. Remarking can be performed in the ingress or the egress side of the hardware pipeline.

The figure below shows how QoS does ingress and egress traffic management


IFP
IFL
Ingress
BA/MF Classification
Ingress Remarking
Policing
Ingress classification based on QoS fields, VLAN priority
Optional remarking of QoS fields in ingress as part of BA classifier
Per Customer 4-level cascaded policer with counters per level (i.e. per class) and color (green, yellow, red).
Egress
Egress Marking
Scheduling / Shaping
Queueing / Shaping
Optional remarking of QoS fields in egress.
Per customer H-QoS scheduler with shaper
Queues to enqueue the subscriber traffic streams (packets) with shaper (with counters per class)
Multi-level Policing (from Subscriber)
BE
LowLoss
LowDelay
VoIP
Cascading policing supported
Multi-level Scheduling and Shaping (towards Subscriber)
VoIP
LowDelay
LowLoss
BE
Subscriber Level
DSLAM/ OLT Level
Port Level

The sections below provide a description of various HQoS components that are configured for both ingress and egress traffic.

- QoS Profiles
- Behavior Aggregate (BA) Classifier
- Multifield (MF) Classifier
- Remarking
- Policer
- Class-Policer-Map
- Queueing
- Traffic Class to Queue Mapping
- Queue-Group
- Scheduler
- Priority Propagation
- Shaper
- Scheduler Mapping
- L2TP QoS
- Multi-level H-QoS : Level-1 to Level-5

QoS Profiles

A profile configuration defines the QoS profile that is attached to either a Subscriber interface or an L3 interface.

Profile maps the following QoS constructs to a Subscriber or an L3 interface:

- Behavioral Aggregate (BA) Classifier
- Multifield (MF) Classifier
- Class Policer Map
- Policer
- Class Queue Map
- Scheduler Map
- Remark Map

Behavior Aggregate (BA) Classifier

Classifiers assign the class to which a packet belongs. BA classification is performed on the ingress and maps incoming packet codepoint to a predefined class. BA Classification relies upon markings (that is, codepoint) placed in the headers of incoming packets:

- IEEE 802.1p: Priority - 3 bits
- IPv4: Type of Service byte (ToS) - 8 bits.
- IPv6: Traffic Class (TC) - 8 bits.
- MPLS: Experimental bits (EXP) - 3 bits.

> - IEEE 802.1p and IPv4/IPv6 classifiers are applied on either Subscriber IFL or L3 IFL by attaching the BA classifier to a profile.
> - MPLS Exp classifiers are applied either globally or per instance (to support multiple VPN marking schemes) by attaching the classifier globally or to an instance.

Classifier configuration has the following guidelines and limitations:

- For IPv4: Only ToS-based classification is possible. Currently, DSCP-based classification is not supported.

- For IPv6: TC-based classification is possible.
- For EXP classification, RBFS uses the uniform mode to copy MSB 3-bits from 8-bits IPv4-ToS or IPv6-TC to the EXP field at the time of MPLS encapsulation at the remote box. IPv4/IPv6 Classifiers do not classify labeled traffic, hence MPLS Classifier is required for the same.

> - Default class for Queue or Policer is *class-0*. If for an incoming packet's *codepoint* there is no class mapping configured under a classifier, the packet will be classified as *class-0*.
> - RBFS supports 8 **classes**: *class-0* to *class-7*.

Ingress Remarking

Ingress remarking is achieved by configuring the "remark-codepoint" field in the Classifier. Ingress remarking rewrites the IPv4-ToS or IPv6-TC field of the incoming packet at the ingress side with configured remark-codepoint. Note that the ingress remarking is not supported for the BA Classifier with match-type MPLS-EXP.

Multifield (MF) Classifier

Multifield (MF) classifiers assign the class to which a packet belongs based on multiple fields. Unlike the BA classifier, where only CoS fields are used for classification, the MF classifier additionally uses the following fields:

- **class**: traffic class of the packet (class-0 to class-7) set by prior BA classifier
- **source prefix**: source IPv4 or IPv6 prefix
- **destination prefix**: destination IPv4 or IPv6 prefix
- **protocol**: UDP or TCP
- **source port**: UDP or TCP source port
- **destination port**: UDP or TCP destination port
- **qos markings**: IPv4 TOS or IPv6 TC header value

The actions supported by a Multifield Classifier are:

- **class**: traffic class to be set (class-0 to class-7)
- **Remark codepoint**: remark codepoint for ingress remarking

> RBFS treats all the incoming IPv4-TOS or IPv6-TC QoS field values in the incoming packet as untrusted. So a user is required to set action-remark-codepoint in the MF Classifier configuration to mark the QoS bits in the IP header of the outgoing packet. If action-remark-codepoint is not configured in the MF Classifier, the default value 0 shall be marked in the packet.

The Multifield Classifiers can be bound globally (global.qos.global.config) or via QoS profile (global.qos.profile.config). The global Multifield Classifier applies to all traffic from any instance or interface. The Multifield Classifier assigned via the QoS profile applies only to ingress traffic received on the interface where the profile is bound to it.

The Multifield Classifier is processed after BA classification which allows it to match on selected class from BA classification or to change the assigned class by more granular match conditions. Both classification stages (BA and MF) are optional, they can be combined or used alone, controlled by configuration.

Multifield Classifiers cannot be bound to MPLS core interfaces. Therefore, the downstream traffic (from core to subscriber) should be classified via global Multifield Classifier, while upstream traffic (from subscriber to core) can be classified via Multifield Classifier from the QoS profile, which is instantiated per subscriber with an implicit match on ingress logical interface (InLIF).

> - RBFS supports 8 **classes**: *class-0* to *class-7*.
> - Per instance MF classifier for MPLS traffic is not supported in RBFS because of hardware limitations.
> - The default class for Queue or Policer is **class-0**. If for an incoming packet, there is no MF classification configured, the packet will be classified as *class-0*.
> - Priority 1 is reserved for BA Classifier ACL entries, therefore the recommendation is to use Priority starting from 2 for MF Classifier
> - If multiple ACL entries are hit in MF having the same priority, the result is unpredictable. So recommendation is to use different priorities for different ACL entries.

Match MPLS traffic

If MF Classifier is to be applied for MPLS traffic (that is, DOWNSTREAM traffic), match MPLS traffic has to be configured in the MF ACL. If not configured, traffic may or may not match the MF ACL entry in the h/w.

Ingress Remarking

Ingress remarking is achieved by configuring the “action remark-codepoint” in the MF Classifier. Ingress remarking rewrites the IPv4-ToS or IPv6-TC field of the incoming packet at the ingress side with configured remark-codepoint.

RADIUS Controlled Dynamic MF Classifier

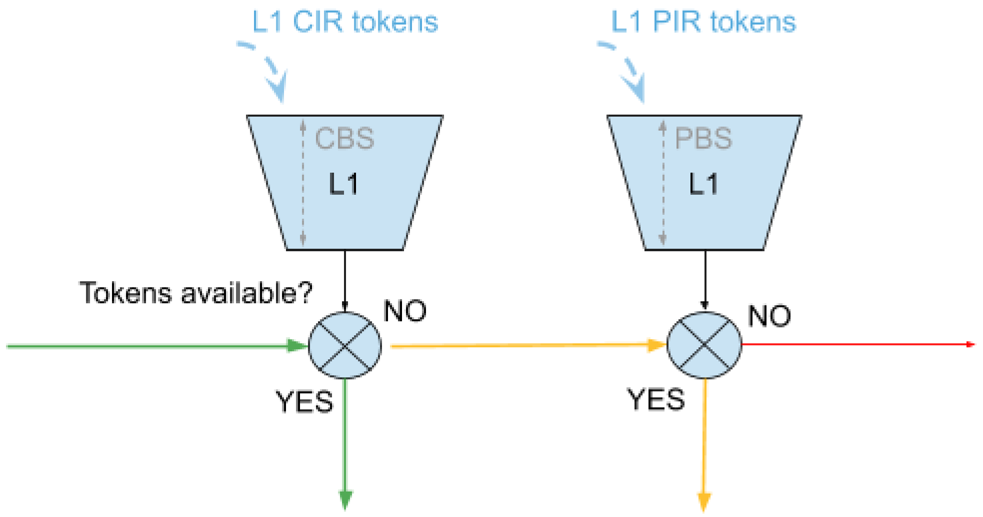
As described in *RBFS RADIUS Services* document dynamic MF Classifier mapping is supported. The dynamic MF Classifier when configured overrides the MF Classifier mapped via QoS profile for the corresponding subscriber but not other subscribers.

For information about configuring the MF Classifier, see Multifield Classifier (MFC) Configuration.

6. Policer

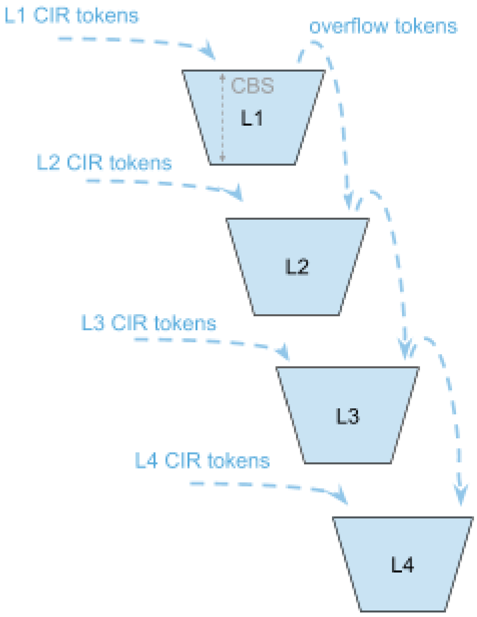
Policer defines the rate at which certain applications can access the hardware resource. So as to rate-limit the traffic from an application, the policer hard-drops the unwanted packets on the ingress side.

In RBFS, policers support the “**two-rate, three-color**” type in a 4-level cascaded mode. This means that each policer level has two rates (CIR and PIR) and three colors (green, yellow and red) with two token buckets as shown below.



This means that traffic below CIR is marked green. Traffic above CIR but below PIR is yellow and above PIR is red. Traffic marked red will be dropped. Traffic marked yellow can be demoted by changing ToS, TC, or EXP using remark map.

In 4-level cascade mode, unused tokens can be passed from higher priority levels to lower priorities where level 1 has the highest and level 4 has the lowest priority as shown in the figure below.



Therefore a lower level configured with CIR 0 can still serve traffic if higher priority levels are not consuming all available tokens.

The available tokens per level are calculated by remaining CIR credits from upper levels and additional credits based on configured CIR per level. Per default, the resulting tokens are not limited. The optional max CIR rate attribute limits tokens from CIR and upper levels. Let us assume levels 1 and 2 are both configured with a CIR of 2Mbps. Without max CIR or max PIR (default behaviour) level 2 can reach up to 4Mbps (level 1 CIR/PIR plus level 2 CIR/PIR). This can be limited by max CIR (for example, 3Mbps). If both level 1 and level 2 have a committed information rate (CIR) of 2Mbps, then level 2 can reach a maximum of 4Mbps (which is the sum of level 1 CIR and level 2 CIR) without any consideration for the maximum CIR. However, the maximum CIR is not relevant for level 1.

Example

	CIR	RX	TX	PIR	RX	TX
L1	2M	1M	1M	2.5M	1M	1M
L2	3M	20M	9M	3.5M / max CIR 8	20M	8M
L3	4M	20M	0M	4.5M	20M	1M
L4	1M	20M	0M	1.5M	20M	0M
SUM	10M	61M	10M	12M	61M	10M

- Here, M indicates Mbps (Megabits per second)

In columns 2 through 4 of the preceding example table, L1 consumes only 1Mbps of the available 4Mbps and passes the remaining 3Mbps to L2 which adds 6m based on their own configured CIR resulting in 9m.

In columns 5 through 7 of the preceding example table, L1 consumes only 1Mbps of the available 4Mbps and passes the remaining 3m to L2 which adds 6Mbps based on their own configured CIR resulting in 9Mbps. But because of the CIR limit set to 8Mbps, only 8Mbps of 9Mbps can be used at this level. The remaining 1Mbps is now passed to L3 which does not add additional CIR-based credits. In both examples, L4 would be able to reach up to 10Mbps if upper levels are not consuming credits.

RADIUS Controlled Dynamic Policer

The RBFS RADIUS services support dynamic policer rate updates. The dynamic policer rate when configured affects only the QoS instance of the corresponding subscriber but not other subscribers.

Class-Policer-Map

Since RBFS supports up to 8 classes but only four policer levels, there is a need to map multiple classes to the same policer level. A *class-policer-map* defines such mappings. Using class-policer-map configuration, one can map any class to any supported policer level (that includes mapping multiple or all classes to the same level). Similar to a policer, a class-policer-map is attached to a profile.

> If class-to-level mapping is not configured, no policing will be applied to the traffic for that class.

7. Queueing

Queuing helps to drop unwanted traffic in advance at the ingress side in case of congestion. This is to ensure bandwidth for qualified services.

RBFS supports the following queueing techniques:

- Tail Drop (TD): This is a conventional congestion avoidance technique. When the network is congested, drop subsequent packets from the queue.
- Weighted Random Early Detection (WRED): This technique requires configuring “Minimum Threshold”, “Maximum Threshold” and “Drop Probability”, which define the start and end range where packets may get discarded. When the average queue size is below the minimum threshold, no packets will be discarded. The drop_probability parameter can be used to specify the drop probability at the max threshold. When the average queue size is between the min and max threshold, the drop probability increases linearly from zero percent (at the min threshold) to drop_probability percent (at the max threshold). When the average queue size is greater than the maximum threshold, all packets are discarded.

  When the average queue size is less than the “Minimum Threshold”, no packets will be discarded.

  When the average queue size is greater than the “Maximum Threshold”, all packets are discarded.

  When the average queue size is between “Minimum Threshold” and “Maximum Threshold”, the drop probability increases linearly from zero percent (at the minimum threshold) to drop probability (at the max threshold).

- Default queue within a queue group is the one mapped to *class-0*. If the classification is not configured for an incoming packet’s codepoint, the packet will be classified as *class-0*. This will be mapped to queue mapped to *class-0* in *Class-Queue-Map*. For more information, see Traffic Class to Queue Mapping.
- Maximum supported Queue size depends upon DRAM/OCB memory. Since OCB is external memory, hardware does not limit the size that can be configured per Queues.

RADIUS Controlled Dynamic Queue

As described for *RBFS RADIUS Services* document dynamic Queue buffer size updates are supported. The dynamic Queue buffer values when configured affect only the QoS instance of the corresponding subscriber but not other subscribers.

Traffic Class to Queue Mapping

The traffic class to queue mapping defines the mapping of classes and queues. The Traffic Class to Queue Mapping is attached to a profile.

- You cannot map two classes to the same queue. The class to queue mapping is 1:1.
- If a queue group is created with four queues, only class-0 to class-3 can be mapped to the queues in class-queue-map; that is, class-4 to class-7 cannot be used.

8. Queue-Group

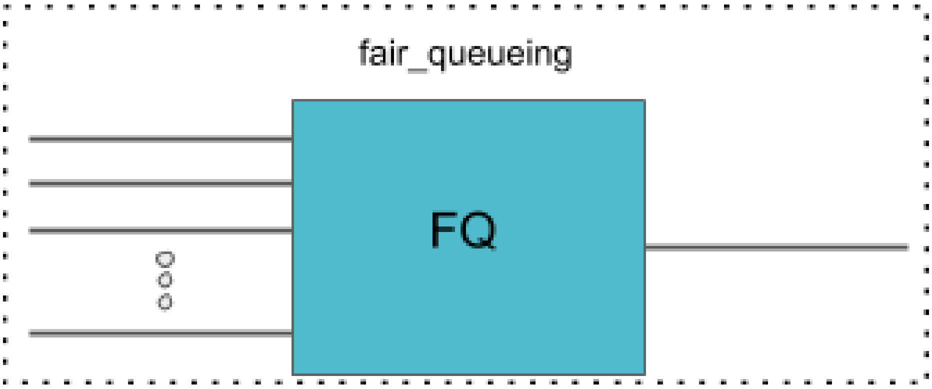
A Queue Group defines the Queue bundle. A Queue Group contains bundle of either 1 or 4 or 8 queues.

9. Scheduler

A scheduler configuration defines scheduler parameters such as type and shaping rate. The shaping rate defined for a scheduler applies to queue(s) associated with it. NOTE: For 1 Queue in Queue-Group, scheduler is not applicable.

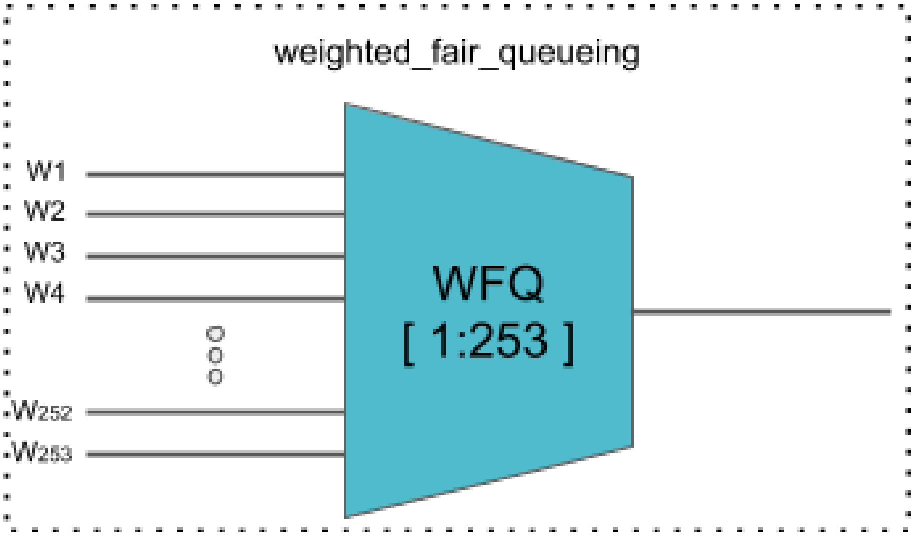
The following scheduler types are supported:

- **Fair Queueing (FQ)**: Uses a round-robin approach to select the next packet to service. This method ensures that all the flows are serviced equally. Configure scheduler type as *fair_queueing* to create FQ scheduler.



- **Weighted Fair Queueing (WFQ)**: Uses a round-robin approach but with no guarantee of flow being serviced equally (like in FQ). The rotation of the next packet to service is based on the weight that is assigned to each flow. Configure scheduler type as *weighted_fair_queueing* to create WFQ scheduler.

  Supported weight: 1 to 253



In any WFQ scheduler, the lower the weight, the higher the

> bandwidth portion is awarded.

- **Strict Priority (SP)**: Uses priority-based approach to service the flow. SP schedulers are supported in “hybrid” mode only. Hybrid mode combines FQ-WFQ schedulers using strict priority.

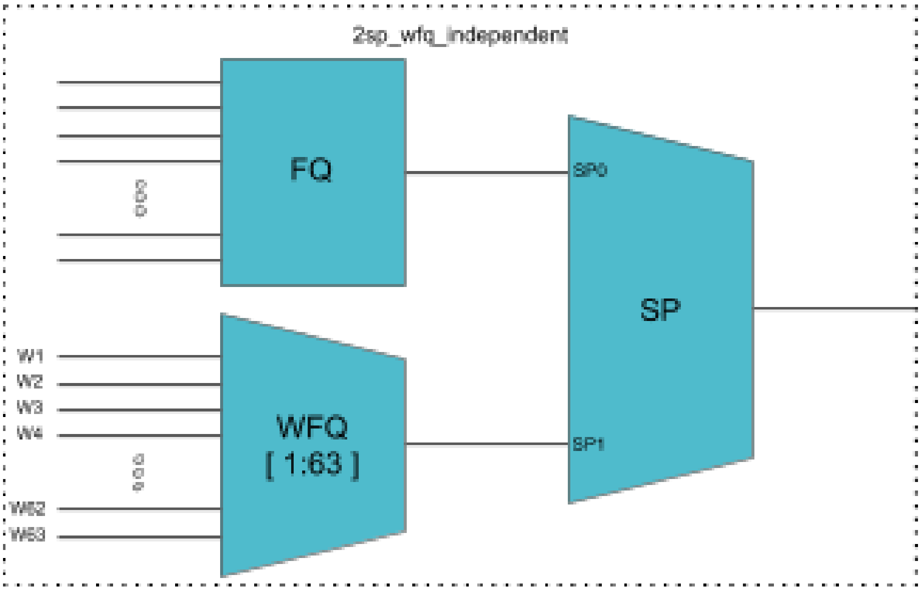
> The priority order for SP is: **strict_priority_0 > strict_priority_1 > strict_priority_2 > strict_priority_3** (where **strict_priority_0** being highest priority and **strict_priority_3** being lowest)

The following SP scheduler types are supported:

- **2 Strict Priority (2SP)**: Uses SP between 1-FQ and 1-WFQ. There are the following types of 2SP hybrid schedulers:

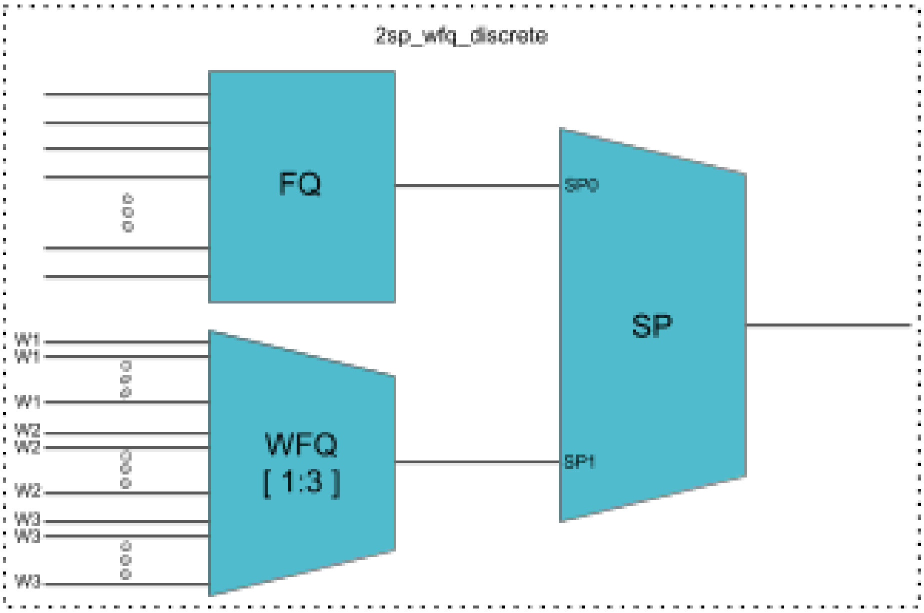
  type **“2sp_wfq_independent”**

  Supported weight: 1 to 63



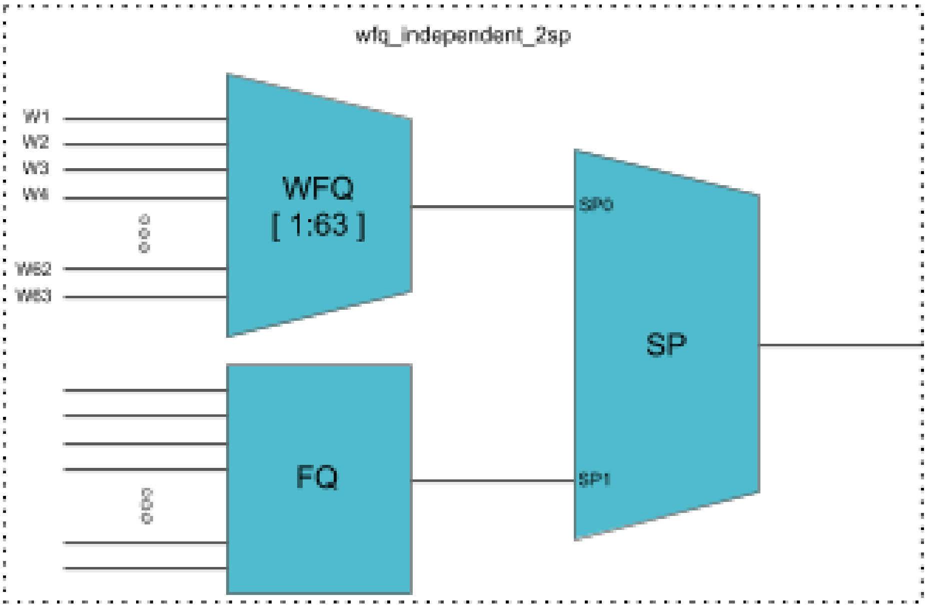
- type **“2sp_wfq_discrete”**

  Supported weight: { 1, 2, 3 }



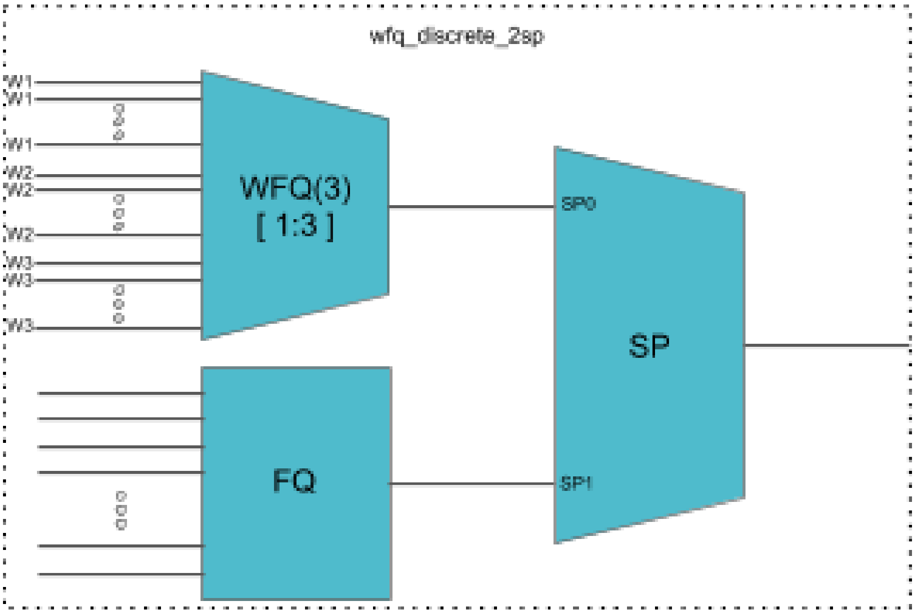
- type **“wfq_independent_2sp”**

  Supported weight: 1 to 63



- type **“wfq_discrete_2sp”**

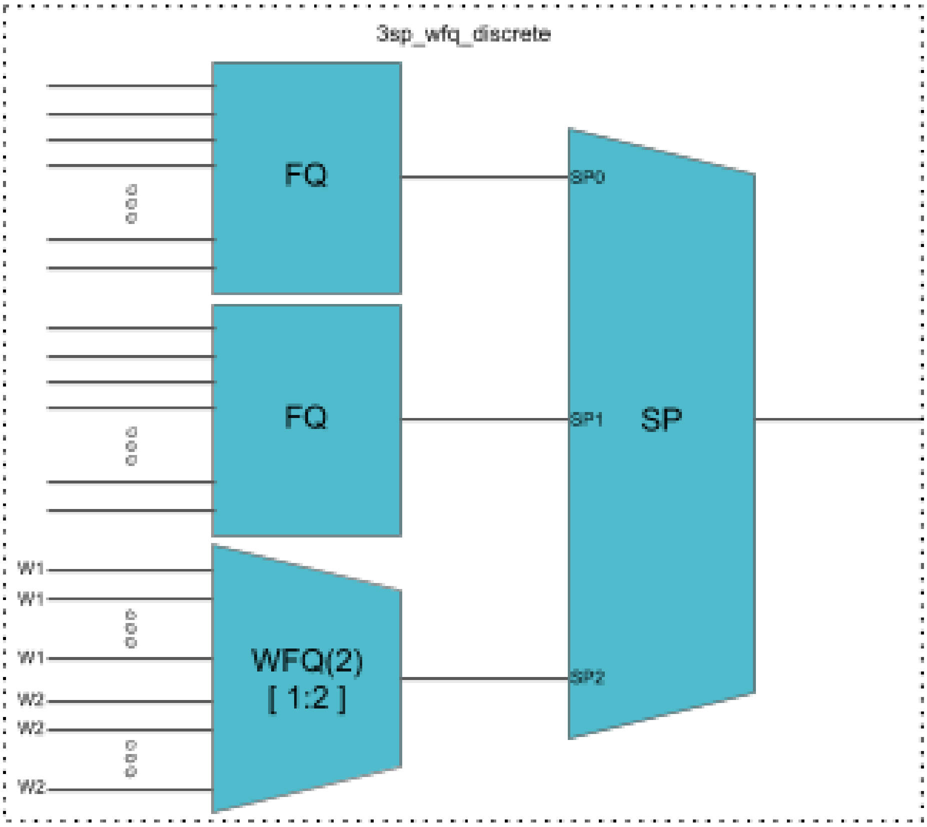
  Supported weight: { 1, 2, 3 }



- **3 Strict Priority (3SP)**: maps 2-FQs and 1-WFQ

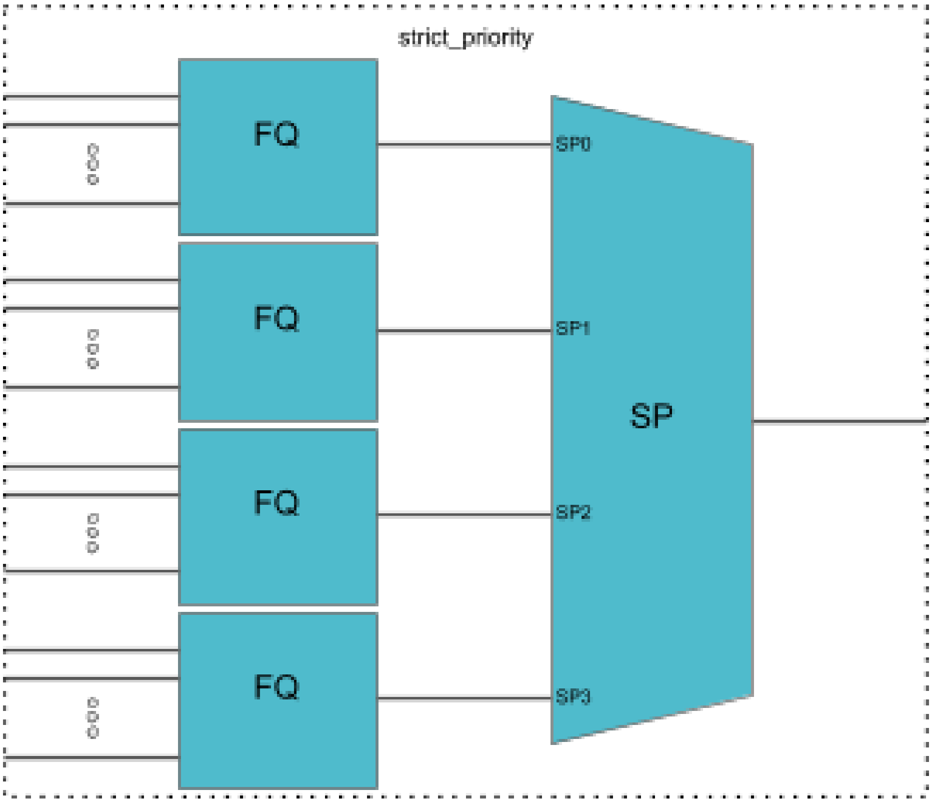
  type: **“3sp_wfq_discrete“**

  Supported weight: { 1, 2 }



- **4 Strict Priority (4SP)**: maps 4-FQs using SP

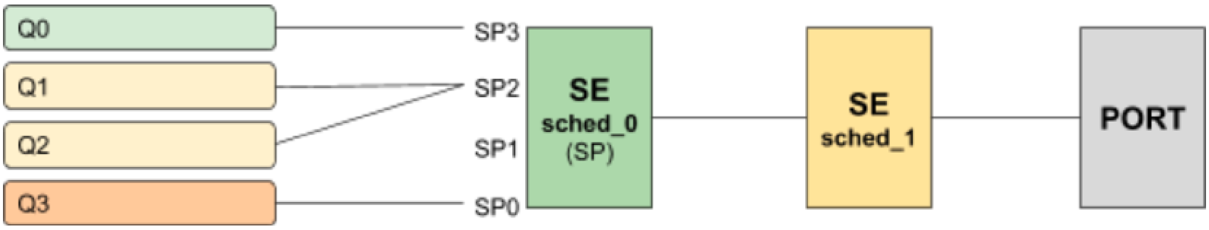
  type **“strict_priority"**



Priority Propagation

Hierarchical QOS (HQoS) on RBFS is implemented by connecting or chaining queues to scheduler elements (Q—> SE), scheduler elements to each other (SE—> SE), and scheduler elements to ports (SE—> PORT). Each scheduler element can have different child connection points based on types described in the section Scheduler.

This means that sched_0 in the example below is not scheduling between the attached queues, but between the different child connection points SP0 to SP3. The scheduler element sched_0 cannot differentiate between Q1 and Q2 in this example because both are connected to SP2.

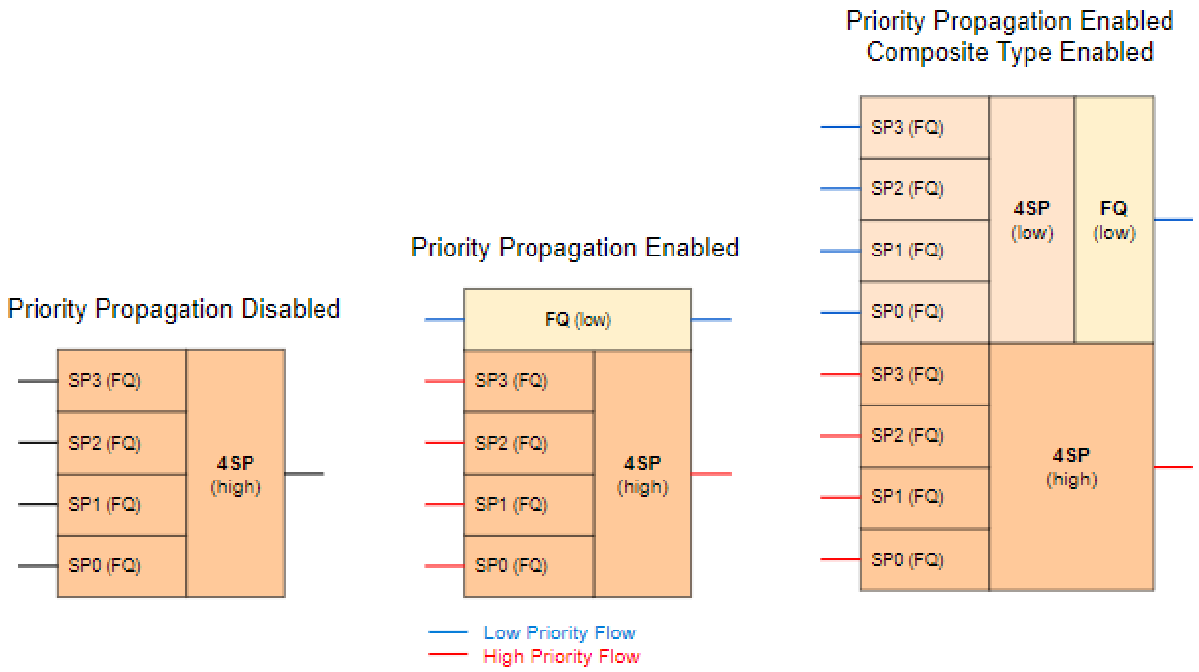


Without priority propagation, each scheduler element can have multiple child connection points but just one parent connection point. Therefore traffic leaving a scheduler element cannot be differentiated by the parent scheduling element. The parent scheduler element sched_1 receives the traffic from sched_0 on the selected child connection point. As already mentioned scheduling within a scheduler element happens between child connection points. Second, a scheduler element has only one parent connection point which can be connected to a child connection point of another scheduler element (output of sched_0 input of sched_1). This results in the situation that all traffic from this SE is handled equally regardless of the queue. This may lead to dropped priority traffic like voice or control traffic in case of congestion in parent elements. For example, if sched_1 has a shaping rate lower than the one of sched_0, it will drop traffic unaware of its original priority.

This problem is addressed with priority propagation which is enabled by default.

With priority propagation, the scheduler elements operate in a dual-flow mode with high and low-priority flows. The credits generated from the physical interface will be consumed by all attached high-priority flows first and only the remaining credits will be available for low-priority flows. In this mode, an implicit FQ element is created for each scheduler element. All queues assigned to low-priority flow will be attached to this element.

An additional composite option of the scheduler element allows also the differentiation between multiple low-priority queues if required. This composite type is created implicitly and does not need to be configured.

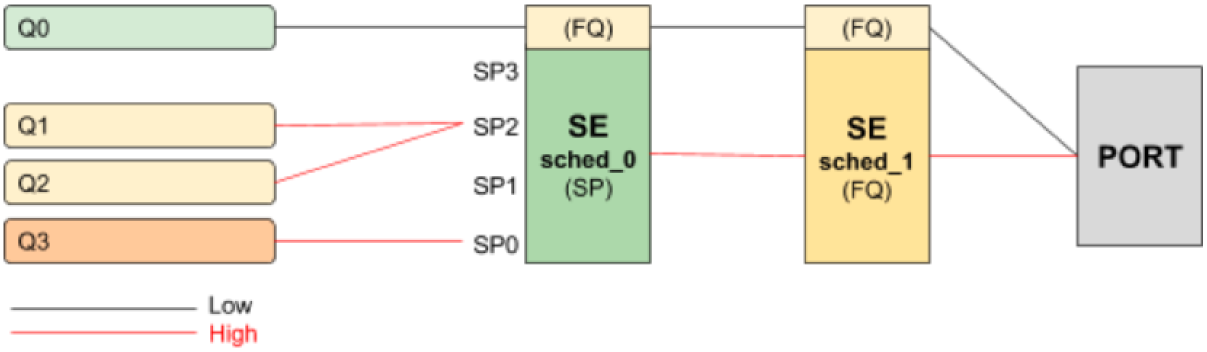


Without priority propagation enabled, each scheduler element consumes only one scheduler resource compared to two elements if enabled. The composite type consumes three scheduler elements.

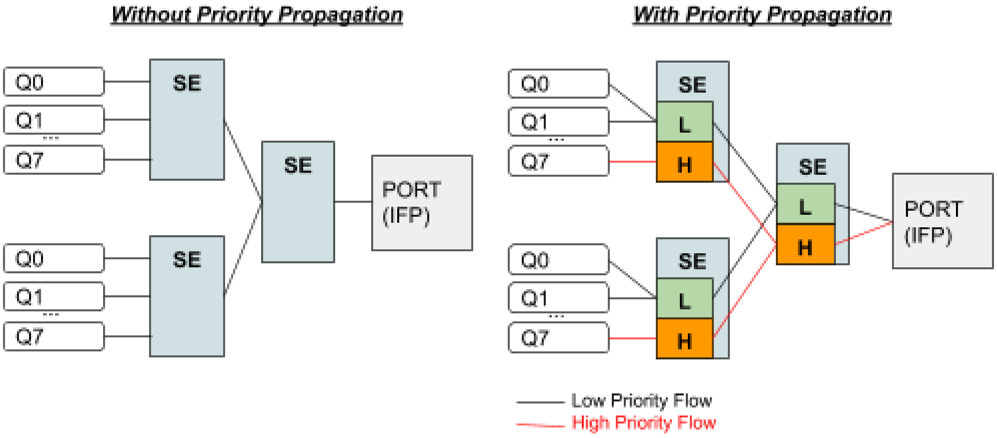
With priority propagation disabled, all traffic is considered a high-priority flow.

Now for each queue, we can select if connected to high-priority or low-priority flow where high-priority flow is selected per default if not explicitly mentioned.

Assuming the example as before but with priority propagation and Q0 assigned to low-priority flow and Q1 - Q3 assigned to high-priority flow.



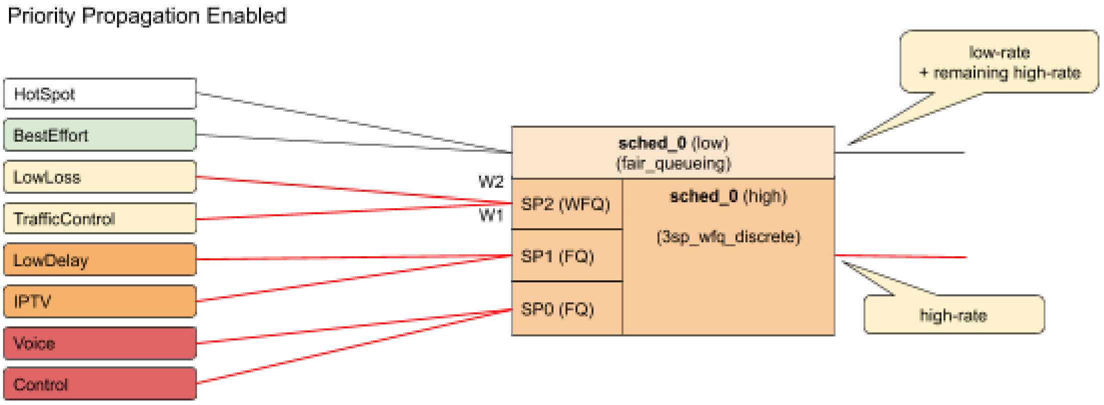
The figure below shows a typical multi-level QoS configuration without priority propagation on the left and with priority propagation on the right side.

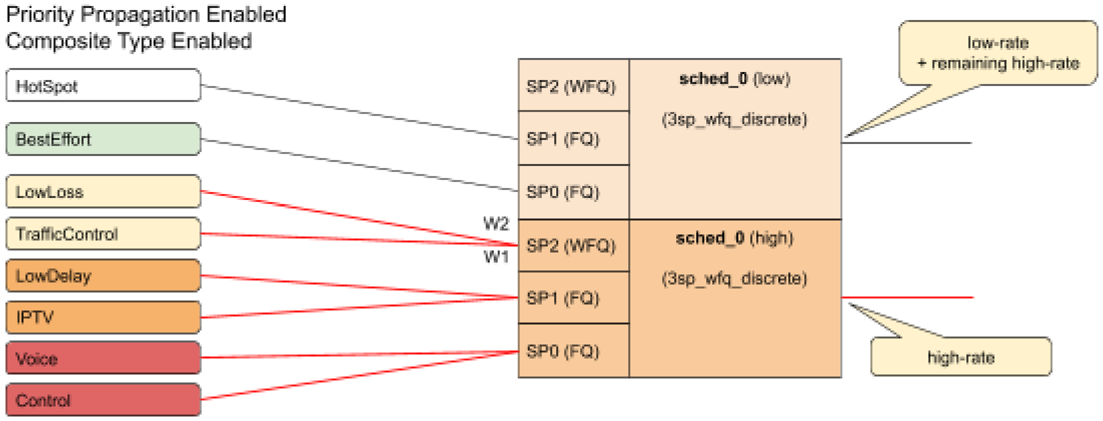


The credits generated from the physical interface will be consumed by high-priority flow first and the remaining credits will be available for low-priority flow. The high-flow traffic at any one element is scheduled based on the type and connection point. Between schedulers, it depends on how they are connected to the parent scheduling element. Per default all levels there is FQ for low and FQ for high-priority flows. The port scheduler is also FQ.

In this mode, each shaper supports two different rates for low and high-priority where the actual shaper rate is the sum of low and high-priority rates. If the low-priority rate is zero, this flow is only served if the high-priority flow is not consuming all credits. An example might be a high rate of 9Mbps and a low rate of 1Mbps which results in 10Mbps for low-priority flow if high-priority flow is not consuming any packets but at least 1Mbps is ensured.

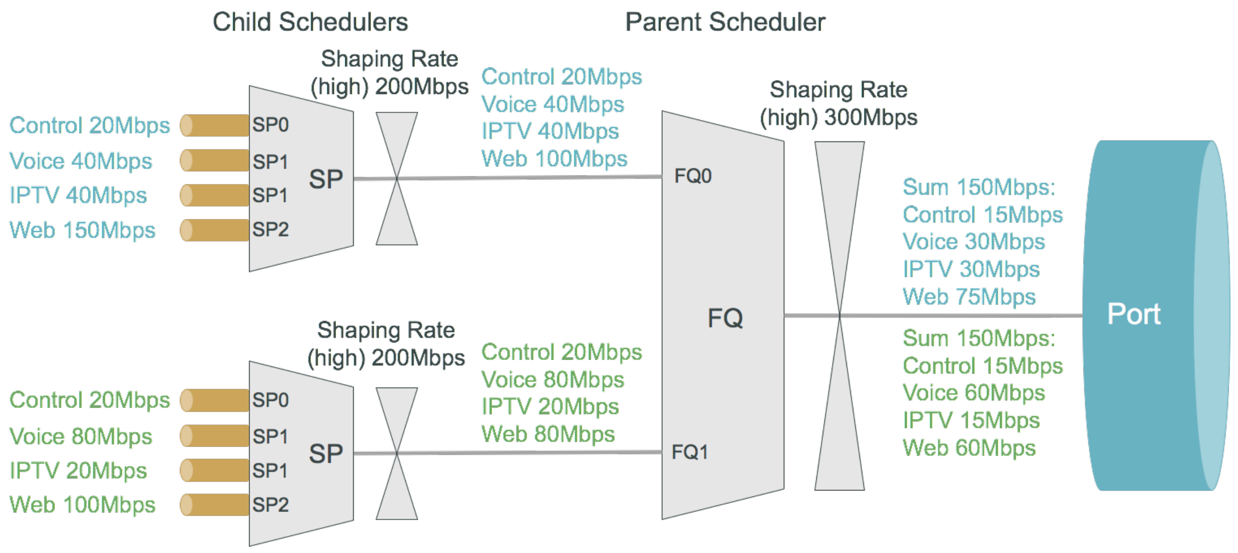
The following example shows a typical access service provider configuration with priority propagation enabled with and without composite type.



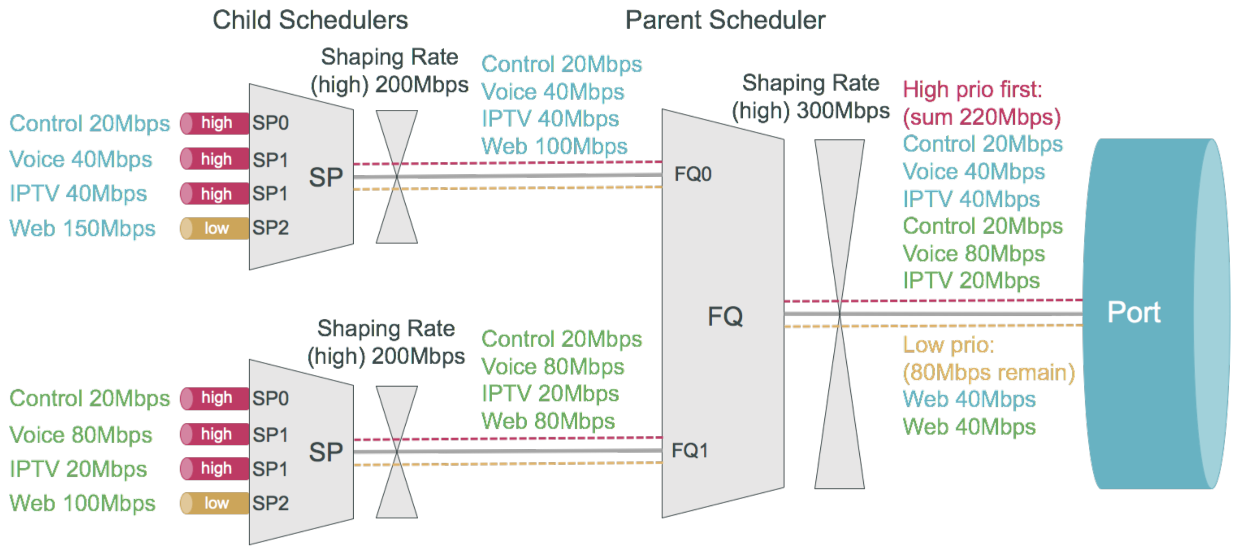


Simple Priority Propagation Scheduling Example

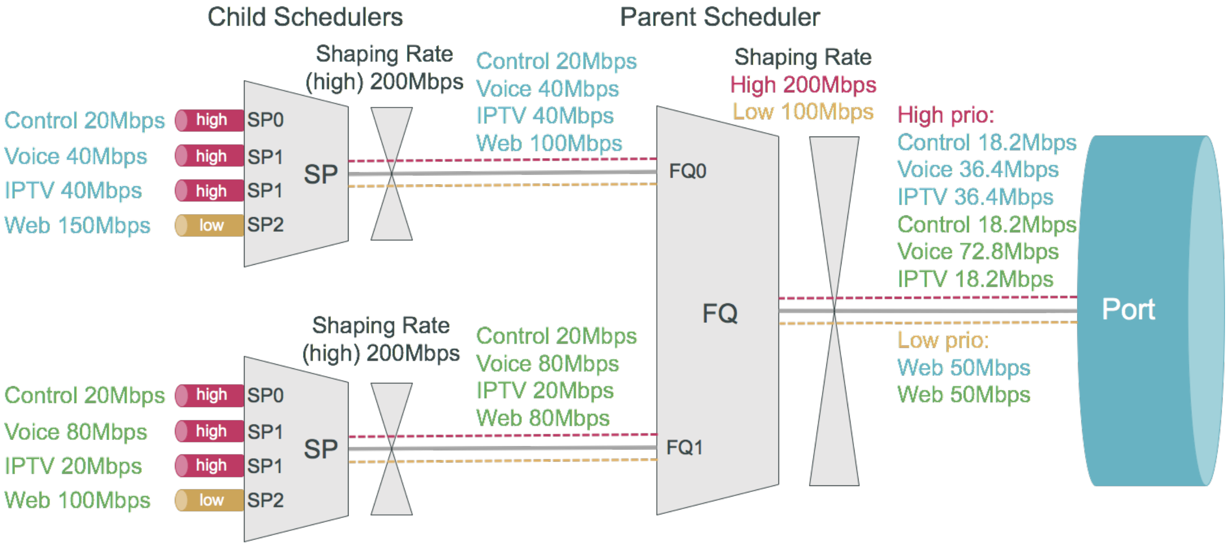
Without priority propagation, the parent scheduler drops traffic equally from all classes as it is unaware of priorities:



With priority propagation, the parent scheduler serves high-priority flows first as shown in the figure below:



With priority propagation and dual-flow shaping, the parent scheduler serves high-priority flows first up to the high-flow shaping rate:


Child Schedulers
Parent Scheduler
Shaping Rate
(high) 200Mbps
Control 20Mbps
Voice 40Mbps
IPTV 40Mbps
Web 100Mbps
Shaping Rate
High 200Mbps
Low 100Mbps
High prio:
Control 18.2Mbps
Voice 36.4Mbps
IPTV 36.4Mbps
Control 18.2Mbps
Voice 72.8Mbps
IPTV 18.2Mbps
Low prio:
Web 50Mbps
Web 50Mbps
Port
Control 20Mbps
Voice 40Mbps
IPTV 40Mbps
Web 150Mbps
high
high
high
low
SP0
SP1
SP1
SP2
SP
FQ0
FQ
FQ1
Shaping Rate
(high) 200Mbps
Control 20Mbps
Voice 80Mbps
IPTV 20Mbps
Web 80Mbps
Control 20Mbps
Voice 80Mbps
IPTV 20Mbps
Web 100Mbps
high
high
high
low
SP0
SP1
SP1
SP2
SP

10. Shaper

Shaper is used to rate-limit the traffic at the egress. In RBFS, shapers can be attached to both Queue and Scheduler.

A shaper configuration defines the shaping rate in Kilo-bits-per-seconds (Kbps).

- Setting the shaping rate to 0 (zero) sets the rate to unlimited. Hence it is recommended to configure at least 1 Kbps so that shaping takes place.
- If shaping rates are to be unconfigured, it needs to be done at all scheduler levels.

10.1. Low-rate Shaping

The Low-rate Shaping feature performs queueing and scheduler-level traffic shaping to rates lesser than 1000 Kbps so that the higher-priority (voice) traffic to flow at optimal levels.

Low-rate Shaping is supported only on high-priority flows, that is high-flow configuration parameter.

RBFS Access-Leaf and Consolidated-BNG platforms have been enabled with Low-rate Shaping by default. For information about the Low-rate Shaping feature enabled platforms, see section 'Feature/Resource Usage" in the *Platform Guide*.

RADIUS Controlled Dynamic Shapers

RBFS RADIUS services support dynamic shaper updates. The dynamic shaper when configured affects only the QoS instance of the corresponding subscriber but not other subscribers.

11. Scheduler Mapping

Scheduler Map defines the set of relationships between parents and children in egress scheduling hierarchy. A child in a Scheduler Map configuration can be either Queue or Scheduler. A parent in a Scheduler Map configuration can be either a Port or Scheduler.

> For 1 Queue in Queue-Group, scheduler-map is not applicable. **Connection Point and Weight**

Child-queue or child-scheduler in a scheduler map configuration is connected to the parent-scheduler at “**connection point (CP)**“. Connection point configuration also has “**weight**” associated with it if the parent has a WFQ scheduler corresponding to that connection point. The valid connection point value for a child to connect to parent **WFQ/FQ** scheduler is **no_priority** and to connect to parent **SP/Hybrid** scheduler is between **strict_priority_0** to **strict_priority_3** (based on a number of Strict Priority points in parent scheduler).

Connection Types

There are five connection types in a scheduler map entry:

- queue_to_port
- queue_to_scheduler
- scheduler_to_scheduler
- scheduler_map_to_scheduler_map
- scheduler_to_port

> - For the **queue_to_port** connection type, the scheduler has no role.

12. Remarking

The packet markers set the codepoint in a packet to a particular value, adding the marked packet to a particular behavior aggregate. When the marker changes the codepoint in a packet, it "remarks" the packet. The codepoint in a packet can be IPv4-ToS, IPv6-TC, MPLS-EXP, or IEEE 802.1p field.

The following remarking options are supported in RBFS:

- IEEE 802.1p : Priority - 3 bits.
- IPv4: Type of Service byte (ToS) - 8 bits.
- IPv6: Traffic Class (TC) - 8 bits.
- MPLS-IPv4: MPLS Experimental bits (EXP) - 3 bits.
- MPLS-IPv6: MPLS Experimental bits (EXP) - 3 bits.

IPv4/v6 and IEEE 802.1p remark-map are applied on an interface - subscriber-ifl or l3ifl using Profile Name.

MPLS-IPv4/v6 remark-map is applied either globally or per instance (to support multiple VPN marking schemes) using Remark-Map Name.

In RBFS, remarking can be performed at the ingress or egress:

- **Ingress remarking** is achieved by configuring the **remark-codepoint** field in the Classifier. Ingress remarking rewrites the IPv4-ToS or IPv6-TC at the ingress side with configured remark-codepoint. The configured remark-codepoint can be modified again at the egress side using remark-map. The ingress remarking is supported for IPv4, IPv6, and IEEE 802.1p BA classifiers. Ingress remarking is supported in MF Classifier as well.
- **Egress remarking** is achieved by configuring the **remark-map**. Remark Map is the mapping of **match-codepoint** and **color** to **remark-codepoint**. Egress remarking helps to remark the IPv4-ToS / IPv6-TC field in the IP header, or to write the EXP field in the MPLS label(s), or to write the IEEE 802.1p field in the VLAN header.

Here *Color* is used to set different *remark-codepoint* for same *match-codepoint* based on color marked by the Policer (i.e. *green* or *yellow*). Color is a mandatory field in remark-map. To set the same *remark-codepoint* for a *match-codepoint* irrespective of color, we have to set color as “*all*”.

IPv4-ToS, IPv6-TC, or MPLS-EXP remarking:

- If the *remark-codepoint* is not configured in the BA Classifier or there is no hit in MF Classifier, match-codepoint in the remark-map is the ToS/TC value of the incoming IP packet.
- If the *remark-codepoint* is configured in the BA Classifier and there is no hit in the MF Classifier, match-codepoint is the same value as the remark-codepoint in the BA Classifier
- Irrespective of the *remark-codepoint* configured in the BA Classifier, if there is a hit in the MF Classifier the *match-codepoint* is the same value as the action remark-codepoint (0 if no action *remark-codepoint* configured) in the MF Classifier.

IEEE 802.1p VLAN remarking:

- Class-to-IP based Remark Map for L2TP UPSTREAM traffic is mapped globally. For more information, see the L2TP QoS section.
- In tunnel termination cases (i.e. Downstream traffic from core to Subscriber) the *remark-codepoint* in the MPLS BA Classifier is of no use. Therefore the *match_codepoint* in remark-map at the egress shall be the ToS/TC value of the incoming IP packet.
- In IP tunnel encapsulation cases (i.e. L2TP Upstream traffic from Subscriber to core) the remark-codepoint in the IPv4-TOS BA Classifier is of no use. Therefore the match_codepoint in class-to-ip remark-map at the egress shall be the Class derived from ingress BA Classifier.
- If no MPLS remarking is configured for the Upstream traffic, EXP bits in the MPLS header are derived from IP header TOS/TC bits using the Uniform MPLS mode.
- For VLAN: Only class to IEEE 802.1p remarking is supported.
- For IPv4: Only ToS based remarking is possible. DSCP-based remarking is not possible.
- For IPv6: Only TC based remarking is possible.

- The VLAN priority remarking support for the platforms is as follows:

  UfiSpace S9600-72XC:

  match-codepoint: Value of the IPv4-ToS/IPv6-TC in the incoming IP packet or the remark-codepoint configured in the BA/MF Classifier

  remark-codepoint: Based on the VLAN priority

  Edgecore AS5916-54XKS:

  match-codepoint: MF or BA classifiers are used to determine class at the ingress

  remark-codepoint: Based on the VLAN priority

  QAX Platform: UfiSpace S9500-22XST and Edgecore CSR320 (AS7316-26XB)

  In QAX platform, IP TOS to VLAN bits do not get directly mapped. The IP ToS is mapped to internal priority in classification, and then from internal priority VLAN priority bit remarking is performed.

- **IPv4/v6 and IEEE 802.1p** remark-map is applied on an interface - subscriber-ifl or l3ifl using Profile Name.
- **MPLS-IPv4/v6** remark-map is applied either globally or per-instance (to support multiple VPN marking schemes) using Remark-Map Name.

L2BSA L2X VLAN Priority Remarking:

The L2BSA L2X VLAN priority remarking support for the platforms is as follows: Only A10NSP switches support VLAN operations, access-leaf is transparent.

- Downstream Traffic (Remote ISP to Subscriber):

  On A10NSP Switch:

  The class derived from the VLAN Pbit Classifier is used to remark MPLS Exp and Outer VLAN IEEE-802.1 Pbit Value, inner VLAN Pbit is transparent.

  On Q2C Access Leaf:

The received Outer VLAN IEEE-802.1 Pbit value is retained, explicit remarking at access-leaf is not supported.

- Upstream Traffic (Subscriber to Remote ISP):

  On Q2C Access Leaf:

  To remark the outgoing packet, global MPLS Exp remark map can be used.

  On A10NSP Switch:

  To remark the outgoing packet, VLAN IEEE-802.1 Pbit remark map can be used.

For more information about L2BSA configurations, see the L2BSA User Guide.

Header Compensation

Queue Compensation

The rate at which the packets are dequeued from a queue depends on the credit received by that queue. The source of the credit received by a queue is the egress port to which the queue is mapped. When a packet is dequeued, the credit balance is decreased by the packet size. But, the packet size that is used must be adjusted to model the packet size at the egress, rather than its actual size at the ingress queue. Thus the header compensation is used to adjust for the differences in header size between ingress queue and egress port. RBFS supports static header compensation configuration per queue (in bytes).

Port Compensation

Similar to queue header compensation where header compensation is performed at the per-queue level, RBFS supports the following header compensation at the per-port level:

- **Ingress Header Compensation**: In line with the header compensation option that we have per-queue, RBFS supports static header compensation configuration at the ingress to be used by the policing. Header compensation changes the effective size of the packet to compensate for changes in header size (such as the CRC removal) when considering the packet for policing. Unlike queue, RBFS ingress header compensation configuration is per ingress port (in bytes).

- **Egress Header Compensation**: In line with the header compensation option that we have per-queue or per-port at the ingress, RBFS supports static header compensation configuration at the egress. The egress header compensation configuration is per egress port (in bytes).

> The supported range for header compensation is -64 to +64 bytes.

L2TP QoS

The Layer 2 Tunneling Protocol (L2TP) QoS for upstream is similar to any other locally terminated subscriber. The QoS Profile is mapped dynamically via RADIUS for the L2TP subscribers.

The L2TP QoS for Downstream requires IPv4-TOS based BA Classifier which is mapped to L2TP Tunnel. The same can be achieved by attaching *l2tp-classifier-name* in *global QoS* configuration.

```
forwarding-options {
    class-of-service {
        global {
            l2tp-classifier-name l2tp-ip;
        }
    }
}
```

For Downstream queueing, there is no change. Queueing is applied using QoS Profile similar to locally terminated Subscribers.

The following features are supported for L2TP QoS.

- Upstream

BA Classifier : IEEE 802.1p

Policing

Policer statistics

Remarking

> - L2TP upstream traffic can be remarked by configuring remark-code-point in the classifier configuration.
> - L2TP downstream traffic can be remarked by profile with the remark configuration. A match-codepoint value will be the

> class value derived from the l2tp-ip classifier.

- Downstream

  BA Classifier : IPv4-TOS

  Queueing/Scheduling/Shaping

  Queue statistics

  Remark-Map : IEEE 802.1p (Class to VLAN priority remarking)

Guidelines

- To avoid control traffic policing/shaping, the assumption is that the IEEE 802.1p bits in Upstream or IPv4-TOS bits in Downstream will be different for control and data traffic, control traffic is expected to have the highest precedence.

  Upstream classification is based on IEEE 802.1p bits.

  Downstream classification is based on IPv4-TOS bits of outer IP header.

- PBIT (IEEE-802.1p) classification and policing for the upstream traffic from IPoE subscribers is not supported.

13. Multi-level H-QoS : Level-1 to Level-5

A sample 5-level requirement looks like this.

Level-1 (IFP)

Physical Interface Shaper.

Level-2 (PON TREE)

Each PON tree is a TDM based shared medium with typically ~2.5 GBit/s (GPON) shared by up to 32 consumers (ONT or DPU).

Level-3 (DPU)

In case of FTTB there is a single DPU with multiple consumers via G.Fast DSL connected which requires an additional hierarchy. This level is not needed for FTTH or FTTC.

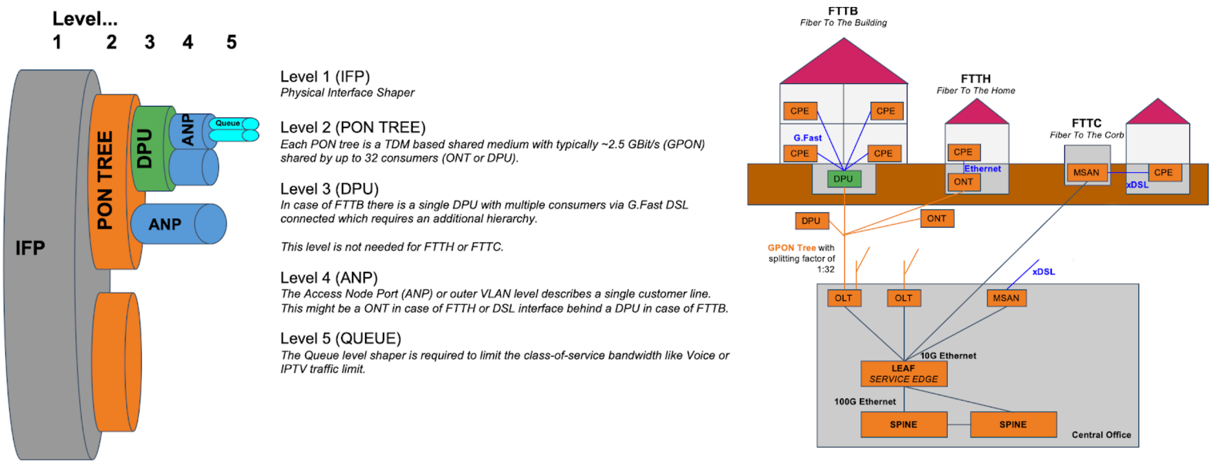
Level-4 (ANP or Session)

The Access Node Port (ANP) or outer VLAN level describes a single customer line. This might be an ONT in case of FTTH or DSL interface behind a DPU in case of FTTB. This level can be also represented on PPPoE sessions as long as just one session is permitted per VLAN.

Level-5 (QUEUE)

The Queue level shaper is required to limit the class-of-service bandwidth like Voice or IPTV traffic limit.

The figure below shows the diagram along with QoS representing Level-1 to Level-5 Hierarchical scheduling.



Level 1 (IFP)
Physical Interface Shaper

Level 2 (PON TREE)
Each PON tree is a TDM based shared medium with typically ~2.5 GBit/s (GPON) shared by up to 32 consumers (ONT or DPU).

Level 3 (DPU)
In case of FTTB there is a single DPU with multiple consumers via G.Fast DSL connected which requires an additional hierarchy.

This level is not needed for FTTH or FTTC.

Level 4 (ANP)
The Access Node Port (ANP) or outer VLAN level describes a single customer line. This might be a ONT in case of FTTH or DSL interface behind a DPU in case of FTTB.

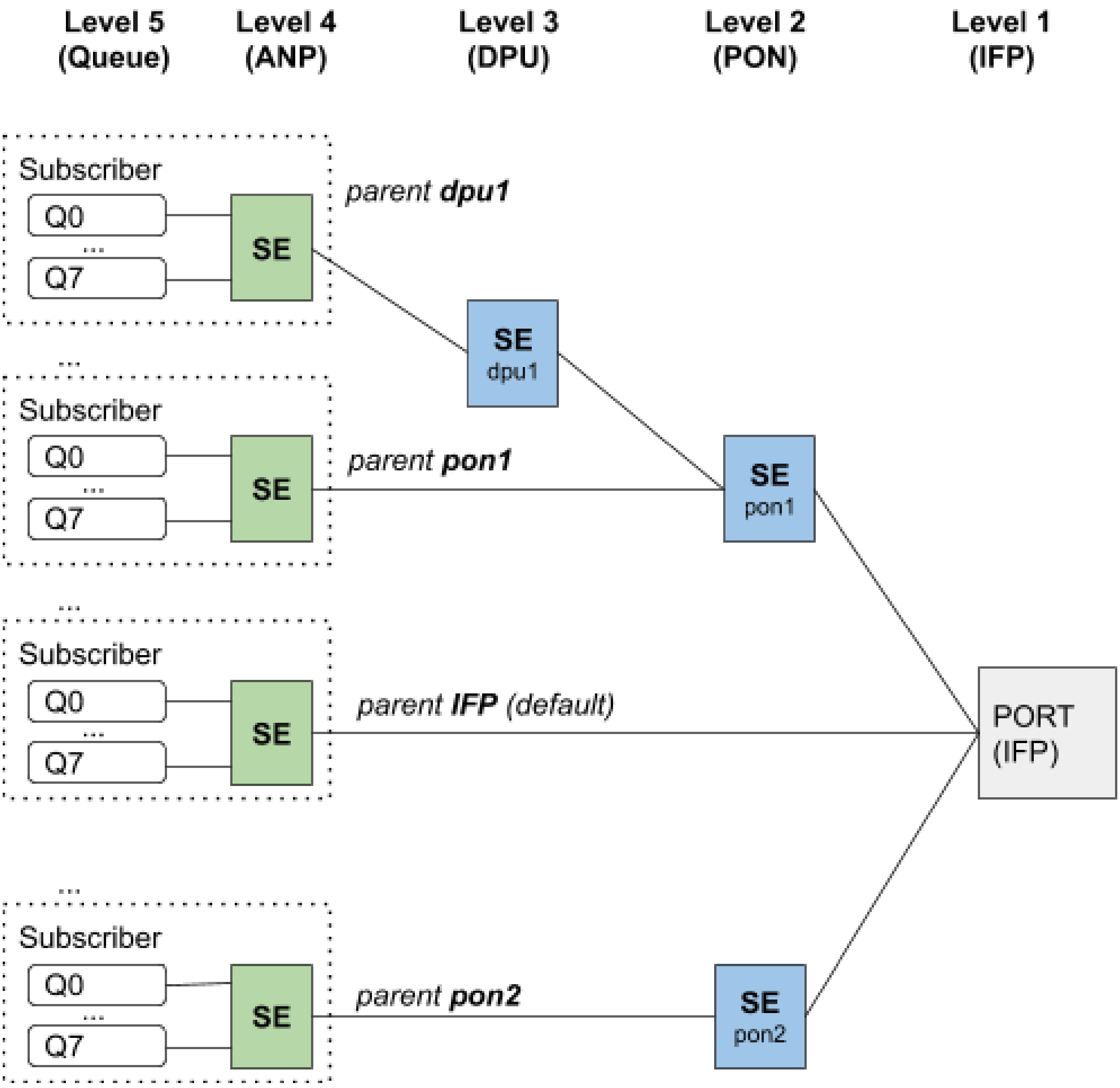
Level 5 (QUEUE)
The Queue level shaper is required to limit the class-of-service bandwidth like Voice or IPTV traffic limit.

The levels 4 and 5 are configured per logical interface (i.e. subscriber-ifl or l3-ifl). Separate scheduler-map representing levels 1 to 3 connectivity shall be statically configured and mapped to the corresponding physical interface (IFP).

The child scheduler in a subscriber scheduler map is connected to the parent scheduler in a physical interface scheduler map using the following way:

- Dynamically via RADIUS in case of dynamic subscribers like PPPoE sessions (Subscriber-IFL).

The figure below shows the same details as the preceding figure before with the different levels but from the DPU-PON-IFP scheduler-map point of view.



MPLS HQoS

The MPLS HQoS has both UNIFORM and PIPE modes. These modes provide the following functionality:

- During MPLS Encapsulation, MPLS Mode is UNIFORM. MSB 3-bits from 8-bits IPv4-ToS or IPv6-TC are copied to the EXP bits of the newly added MPLS header(s).
- During MPLS Decapsulation, MPLS Mode is PIPE. 8-bit IPv4-ToS or IPv6-TC will be retained and hence it provides ToS/TC codepoint transparency.

For the Uniform MPLS mode mapping between IPv4-ToS or IPv6-TC to MPLS-EXP see the table below:

IPv4-TOS / IPv6-TC	EXP	DSCP
0-31	0	0-7
32-63	1	8-15
64-95	2	16-23
96-127	3	24-31
128-159	4	32-39
160-191	5	40-47
192-223	6	48-55
224-255	7	56-63

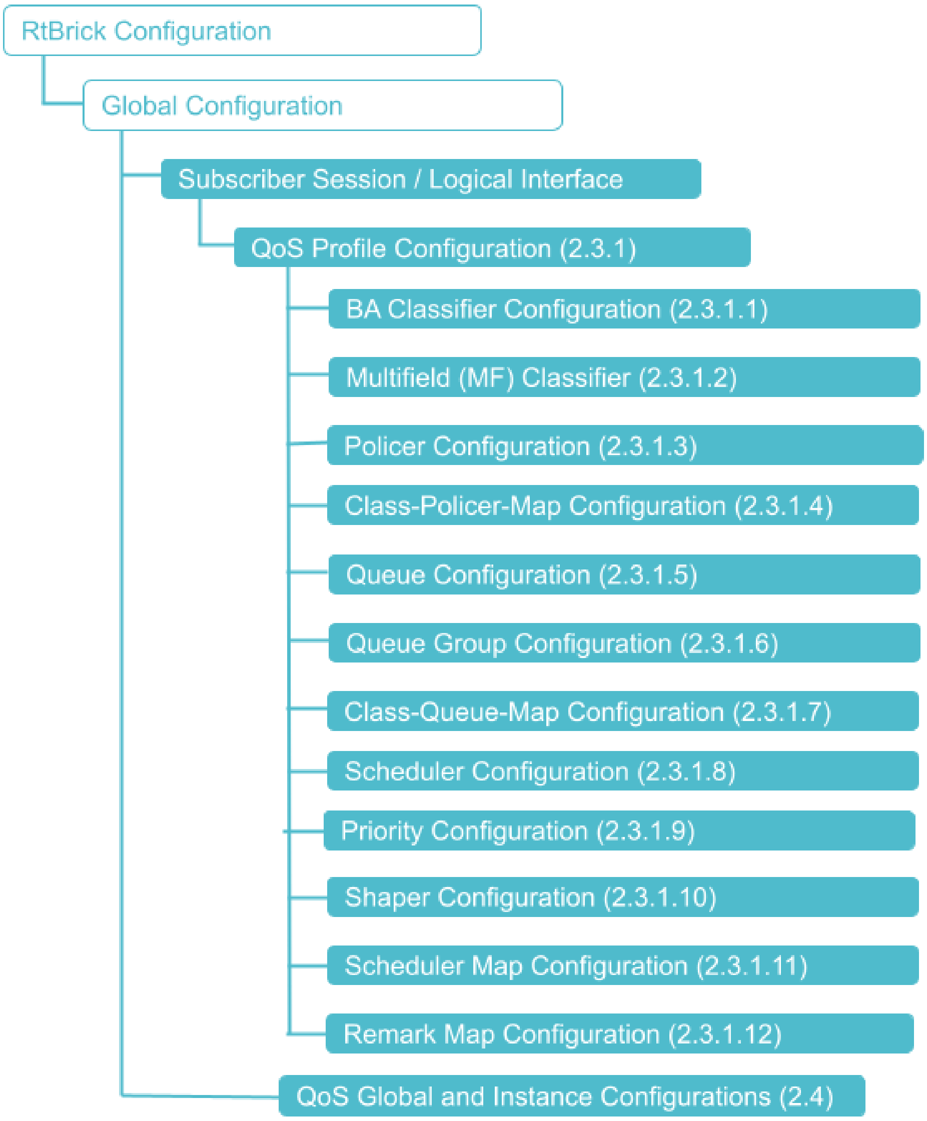
Supported Platforms

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

13.3. HQoS Configuration

Basic QoS Configuration Elements

The figure below shows the dependencies between the various QoS configuration elements.


RtBrick Configuration
Global Configuration
Subscriber Session / Logical Interface
QoS Profile Configuration (2.3.1)
BA Classifier Configuration (2.3.1.1)
Multifield (MF) Classifier (2.3.1.2)
Policer Configuration (2.3.1.3)
Class-Policer-Map Configuration (2.3.1.4)
Queue Configuration (2.3.1.5)
Queue Group Configuration (2.3.1.6)
Class-Queue-Map Configuration (2.3.1.7)
Scheduler Configuration (2.3.1.8)
Priority Configuration (2.3.1.9)
Shaper Configuration (2.3.1.10)
Scheduler Map Configuration (2.3.1.11)
Remark Map Configuration (2.3.1.12)
QoS Global and Instance Configurations (2.4)

To configure QoS, perform the following steps which include creating a QoS profile and enabling QoS on a Subscriber-Interface or L3-Interface.

1. Define a QoS profile with classifier, multifield-classifier, class-policer-map, policer, class-queue-map, scheduler-map, and remark-map based on SLAs.
2. Create a Behavioral Aggregate (BA) and/or Multifield (MF) classifier to classify the network traffic at the ingress.
3. Create a policer to police the classified traffic at the ingress with the CIR/PIR defined.
4. Create necessary class-to-policer-map to map the classes to policer levels (mandatory for policing).
5. Create necessary queues with proper size to queue the classified traffic before egressing the traffic.

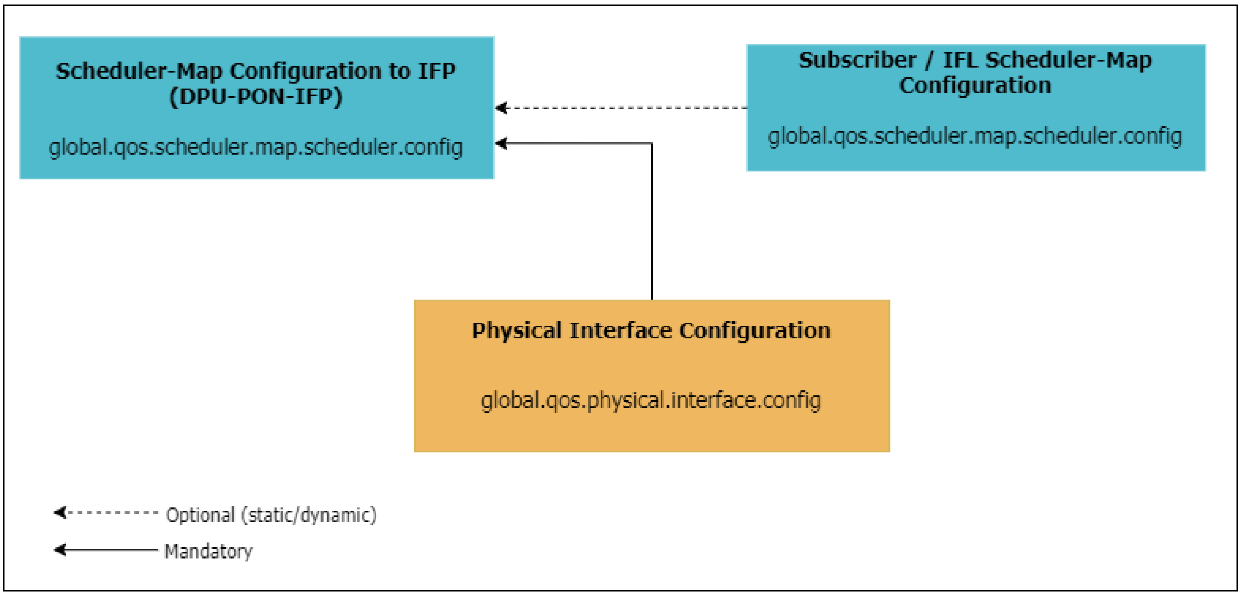
6. Create queue groups and define queue numbers (1/4/8).
   NOTE: For the system to initialize with a single queue in a Queue-Group, a system reboot is required.
7. Create the necessary class-to-queue map to map the classes to queues (mandatory for queuing).
8. Specify scheduler(s) with type as required.
9. (Optional) Create the shaper (low-flow / high-flow) and attach it to queue(s) and/or scheduler(s).
10. Specify a scheduler map to define a set of relationships between parent (scheduler or port) and child (queue/queue-group or scheduler) at the egress.
11. (optional) Create Remark-Map for QoS field remarking of the outgoing packet.
12. For the downstream traffic, map the MPLS EXP classifier either to an instance or global entity and/or the Multifield (MF) classifier as a global entity (refer to the figure above.)
13. (optional) Map the MPLS-IPv4/IPv6 remark-map either to an instance or configure it as a global entity.

> Priority propagation is enabled by default. To disable the Priority Propagation, we recommend doing this at the beginning and not during an active session.

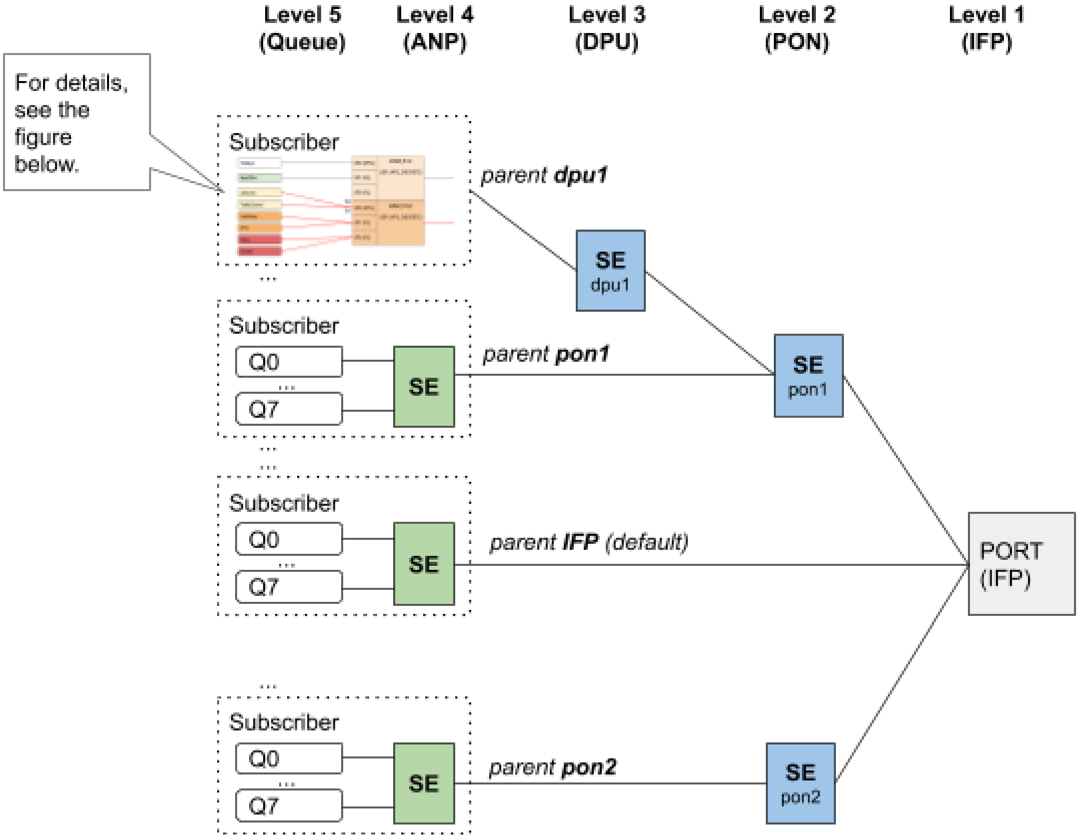
Hierarchical QoS

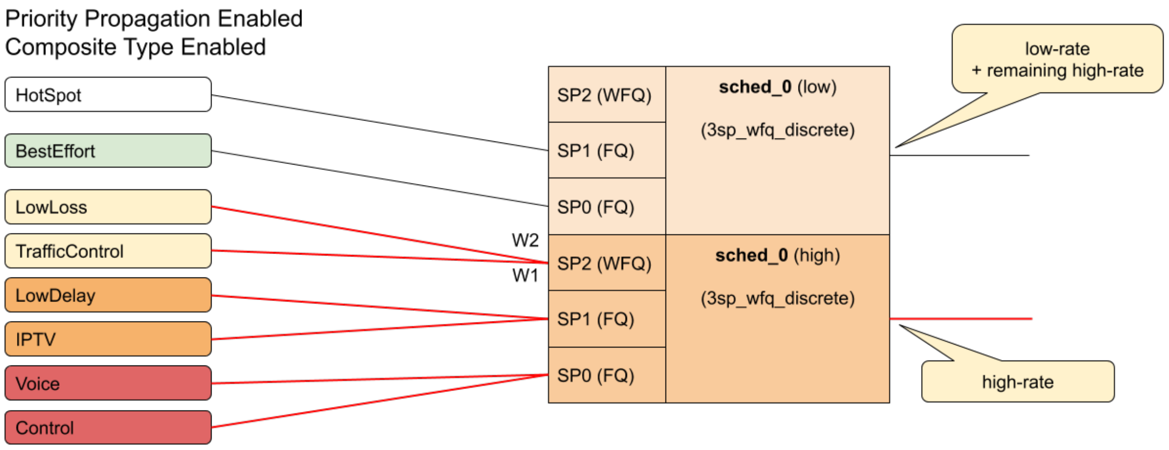
Basic Quality of Service (QoS) is designed to provide a single level of traffic scheduling. In contrast, Hierarchical Quality of Service (HQoS) offers a more sophisticated approach to traffic treatment, utilizing multiple levels of scheduling in a hierarchical manner based on Service Level Agreements (SLAs).

The figure below shows the additional dependencies for Multi-level HQoS.


Scheduler-Map Configuration to IFP (DPU-PON-IFP)
global.qos.scheduler.map.scheduler.config
Subscriber / IFL Scheduler-Map Configuration
global.qos.scheduler.map.scheduler.config
Physical Interface Configuration
global.qos.physical.interface.config
Optional (static/dynamic)
Mandatory

Below are the figures depicting an example of scheduling hierarchy.





To configure HQoS, map the level-3 to level-5 hierarchy in multi-level HQoS to a different scheduler to represent it and map it to the physical interface.

Configuration Syntax and Commands

The configurations in this section exemplify the setup of Hierarchical Quality of Service (HQoS) for a subscriber with a residential profile and a Service Level Agreement (SLA) of 20Mbps upstream and downstream. The setup also encompasses four different service types, namely Best Effort(BE), Low-Delay(LD), Low-Loss(LL) and Voice(VO). Different services are assigned to various policer levels for upstream traffic based on their respective traffic classes. Meanwhile, for downstream traffic, these same services are assigned to different queues depending on their traffic class. The traffic class is determined by the configuration of either the BA or MF classifier.

The following sections describe the HQoS configuration syntax and commands.

- Configuring QoS Profiles
- Behavior Aggregate (BA) Classifier Configuration
- Multifield Classifier (MFC) Configuration
- Policer Configuration
- Class-Policer-Map Configuration
- Queue Configuration
- Queue-Group Configuration
- Class-Queue-Map Configuration

- Scheduler Configuration
- Shaper Configuration
- Scheduler-Map Configuration
- Remark-Map Configuration

13.4. Configuring QoS Profiles

A profile configuration defines the QoS profile that is installed on the subscriber or L3 interfaces. The following sections explain the different elements of the QoS profiles that you can create and attach.

Use the following CLI syntax to configure a QoS profile:

```
set forwarding-options class-of-service profile <profile-name> <attribute> <value>
```

Attribute	Description
<profile-name>	Specifies the name of the QoS profile
classifier-name <classifier-name>	Specifies the name of the BA classifier
multifield-classifier-name <multifield-classifier-name>	Specifies the name of the multifield classifier
class-policer-map-name <class-policer-map-name>	Specifies the name of the map that associates a class with a policer level.
policer-name <policer-name>	Specifies the policer name
class-queue-map-name <class-queue-map-name>	Specifies the name of the map that associates a class with a queue
scheduler-map-name <scheduler-map-name>	Specifies the name of the scheduler map
remark-map-name <remark-map-name>	Specifies the name of the remark map

Attribute	Description
egress-class-policer-map-name <egress-class-policer-map-name>	Specifies the name of the egress Class Policer Map. This configuration applies only to the L2X traffic on the LAG interface.
egress-policer-name <egress-policer-name>	Specifies the name of the egress policer. This configuration applies only to the L2X traffic on the LAG interface.

In the following example, the QoS profile residential is configured with classifier-name subs-pbit-class, class-policer-map policer-map-residential, policer policer-residential, class-queue-map subs-4queues, scheduler-map subs-4queues-residential, and remark-map subs-remarking-residential.

```
set forwarding-options class-of-service profile residential
set forwarding-options class-of-service profile residential classifier-name subs-pbit-class
set forwarding-options class-of-service profile residential class-queue-map-name subs-4queues
set forwarding-options class-of-service profile residential remark-map-name subs-remarking-residential
set forwarding-options class-of-service profile residential policer-name policer-residential
set forwarding-options class-of-service profile residential class-policer-map-name policer-map-residential
set forwarding-options class-of-service profile residential scheduler-map-name subs-4queues-residential
commit
```

The following example shows the QoS profile configuration of the residential profile:

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service profile
{
  "rtbrick-config:profile": [
    {
      "profile-name": "residential",
      "classifier-name": "subs-pbit-class",
      "class-queue-map-name": "subs-4queues",
      "remark-map-name": "subs-remarking-residential",
      "policer-name": "policer-residential",
      "class-policer-map-name": "policer-map-residential",
      "scheduler-map-name": "subs-4queues-residential"
    }
  ]
}
supervisor@rtbrick>LEAF01: cfg>
```

Behavior Aggregate (BA) Classifier Configuration

Use the following CLI syntax to configure the BA classifier:

set forwarding-options class-of-service classifier <classifier-name>

```
<attribute> <value>
```

Attribute	Description
<classifier-name>	Specifies the classifier name
match-type <match-type>	Specifies the type of traffic to classify, that is, ipv4-tos, ipv6-tc, ieee-802.1, exp
match-type <match-type> codepoint <codepoint>	Specifies the code-point value based on the match-type
match-type <match-type> codepoint <codepoint> class <class>	Specifies the traffic class as class-0, class-1, class-2, class-3, class-4, class-5, class-6, and class-7
match-type <match-type> codepoint <codepoint> remark-codepoint <remark-codepoint>	Specifies the remark-codepoint that is used for remarking

The following example configures the BA classifier subs-pbit-class for traffic ieee-802.1 with a match code point 2 classified as class class-1.

```
set forwarding-options class-of-service classifier subs-pbit-class match-type ieee-802.1 codepoint 2 class
class-1
commit
```

The following example shows BA classifier configuration.

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service classifier
{
  "rtbrick-config:classifier": [
    {
      "classifier-name": "subs-pbit-class",
      "match-type": [
        {
          "match-type": "ieee-802.1",
          "codepoint": [
            {
              "codepoint": 2,
              "class": "class-1"
            }
          ]
        }
      ]
    }
  ]
}
supervisor@rtbrick>LEAF01: cfg>
```

Multifield Classifier (MFC) Configuration

Use the following CLI syntax to configure the multifield classifier.

```
set forwarding-options class-of-service multifield-classifier <attribute>
<value>
```

Starting from Release 20.10.2, Multifield Classifier configuration requires explicit use of ordinal keywords.

Attribute	Description
acl <l3v4 | l3v6> <…>	Specifies the l3v4 / l3v6 ACL rule for multifield classifier configurations. For more information on configuring the ACL match rules, see the sections below.

IPv4 ACL Configuration Configuration

Use the following CLI syntax to configure the IPv4 ACL Match Configuration for the multifield classifier:

```
set forwarding-options class-of-service multifield-classifier acl l3v4 rule
<rule-name> ordinal <ordinal-value> <attribute> <Value>
```

Attribute	Description
<rule-name>	Specifies the multifield classifier rule name
ordinal <ordinal-value>	Specifies the ordinal that is used for traffic policy rule referencing
match <destination-ipv4-prefix>	Specifies the destination IPv4 prefix address
match <destination-ipv4-prefix-list>	Specifies the destination IPv4 prefix list name
match <direction>	Specifies the acl l3 traffic direction match
match <ip-protocol>	Specifies the IP protocol such as UDP or TCP

Attribute	Description
match <destination-l4-port>	Specifies the Layer 4 destination port number
match <ipv4-tos>	Specifies the IPv4 ToS value
match <ipv4-dscp>	Specifies the IPv4 Differentiated Services Code Point (DSCP) value
match <forward-class>	Specifies the forward class name
match <mpls-traffic>	Specifies the MPLS traffic
match <source-ipv4-prefix>	Specifies the source IPv4 prefix address
match <source-ipv4-prefix-list>	Specifies the source IPv4 prefix list name
match <source-l4-port>	Specifies the Layer 4 source port number
action forward-class <class>	class-0, class-1, class-2, class-3, class-4, class-5, class-6, class-7
action remark-codepoint <remark-codepoint>	Specifies the remark-map codepoint value

Multifield classification commonly matches on IP address fields, the IP protocol type field, or the port number in the UDP or TCP pseudo-header field.

The l3v4 acl-type multifield classifier is configured with a qualifier/match based on ipv4-tos (128, 160) and source-ipv4-prefix (132.1.1.3/32) with action as forward-class (class-0,class-1).

```
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc ordinal 1001
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc ordinal 1001 match
ipv4-tos 128
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc ordinal 1001 match
source-ipv4-prefix 132.1.1.3/32
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc ordinal 1001 action
forward-class class-0
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc ordinal 1002
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc ordinal 1002 match
ipv4-tos 160
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc ordinal 1002 match
source-ipv4-prefix 132.1.1.3/32
set forwarding-options class-of-service multifield-classifier acl l3v4 rule global_mfc ordinal 1002 action
forward-class class-1
commit
```

The following example shows the IPv4 Match Configuration for the multifield classifier:

```
supervisor@rtbrick>C-BNG.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"
            }
          }
        ]
      }
    ]
  }
```

IPv6 ACL Configuration Configuration

Use the following CLI syntax to configure the IPv6 ACL match configuration for the multifield classifier:

set forwarding-options class-of-service multifield-classifier acl l3v6 rule <rule-name> ordinal <ordinal-value> <attribute> <value>

Attribute	Description
<rule-name>	Specifies the multifield classifier rule name
ordinal <ordinal-value>	Specifies the ordinal that is used for traffic policy rule referencing
match <destination-ipv6-prefix>	Specifies the destination IPv6 prefix address
match <destination-ipv6-prefix-list>	Specifies the destination IPv6 prefix list name
match <ip-protocol>	Specifies the IP protocol such as UDP or TCP

Attribute	Description
match <destination-l4-port>	Specifies the Layer 4 destination port number
match <ipv6-tc>	Specifies the IPv6 traffic class value
match <forward-class>	Specifies the forward class name
match <mpls-traffic>	Specifies the MPLS traffic
match <source-ipv6-prefix>	Specifies the source IPv6 prefix address
match <source-ipv6-prefix-list>	Specifies the source IPv6 prefix list name
match <source-l4-port>	Specifies the Layer 4 source port number
action forward-class <class>	class-0, class-1, class-2, class-3, class-4, class-5, class-6, class-7
action remark-codepoint <remark-codepoint>	Specifies the remark-map codepoint value

In the following example, the l3v6 acl-type multifield classifier is configured with qualifier/match based on ipv6-tc (128, 160) and source-ipv6-prefix (132:1:1::3/32) with action as forward-class (class-0,class-1).

```
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1001
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1001 match
ipv6-tc 128
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1001 match
source-ipv6-prefix 132::1::3/32
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1001 action
forward-class class-0
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1002
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1002 match
ipv6-tc 160
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1002 match
source-ipv6-prefix 132:1:1::3/32
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1002 action
forward-class class-1
commit
```

The following example shows the IPv6 match configuration for the multifield classifier:

```
supervisor@rtbrick>C-BNG.rtbrick.net: cfg> show config forwarding-options class-of-service multifield-
classifier acl l3v6 rule global_mfc
{
    "rtbrick-config:rule": [
      {
        "rule-name": "global_mfc",
        "ordinal": [
          {
            "ordinal-value": 1001,
            "match": {
              "ipv6-tc": 128,
              "source-ipv6-prefix": "132:1:1::3/32"
            },
            "action": {
              "forward-class": "class-0"
            }
          },
          {
            "ordinal-value": 1002,
            "match": {
              "ipv6-tc": 160,
              "source-ipv6-prefix": "132:1:1::3/32"
            },
            "action": {
              "forward-class": "class-1"
            }
          }
        ]
      }
    ]
  }
```

IPv4/IPv6 Priority Configuration

Use the following CLI syntax to configure the IPv4/IPv6 priority configuration:

set forwarding-options class-of-service multifield-classifier acl [**l3v4** |**l3v6**] **rule** <rule-name> <attribute> <value>

Attribute	Description
<rule-name>	Specifies the multifield classifier rule name
ordinal <ordinal-value>	Specifies the ordinal that is used for traffic policy rule referencing
ordinal <ordinal-value> <priority>	Specify the ACL priority value. Range: 0 - 65535.

The following example configures the l3v6 acl-type multifield classifier with qualifier/match based on ipv6-tc (128) and source-ipv6-prefix(132:1:1::3/32) with action as forward-class (class-0,class-1) along with priority value (100, 200). When there are multiple qualifiers or actions, the one with the higher priority takes precedence over the others.

```
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1001
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1001 match
ipv6-tc 128
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1001 match
source-ipv6-prefix 132::1::3/32
```

```
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1001 priority
100
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1001 action
forward-class class-0
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1002
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1002 match
ipv6-tc 128
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1002 match
source-ipv6-prefix 132:1:1::3/32
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1002 priority
200
set forwarding-options class-of-service multifield-classifier acl l3v6 rule global_mfc ordinal 1002 action
forward-class class-1
commit
```

The following example shows the IPv4 priority configuration:

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service multifield-classifier acl
l3v4 rule rtb_mfc


supervisor@rtbrick>C-BNG.rtbrick.net: cfg> show config forwarding-options class-of-service multifield-
classifier acl l3v6 rule global_mfc
{
    "rtbrick-config:rule": [
      {
        "rule-name": "global_mfc",
        "ordinal": [
          {
            "ordinal-value": 1001,
            "priority": 100,
            "match": {
              "ipv6-tc": 128,
              "source-ipv6-prefix": "132:1:1::3/32"
            },
            "action": {
              "forward-class": "class-0"
            }
          },
          {
            "ordinal-value": 1002,
            "priority": 200,
            "match": {
              "ipv6-tc": 160,
              "source-ipv6-prefix": "132:1:1::3/32"
            },
            "action": {
              "forward-class": "class-1"
            }
          }
        ]
      }
    ]
  }
supervisor@rtbrick>LEAF01: cfg>
```

13.4.1. Policer Configuration

Use the following CLI syntax to configure the QoS policer:

set forwarding-options class-of-service policer <policer-name> <attribue> <value>

Attribute	Description
<policer-name>	Specifies the policer name.
<levels>	Specifies levels in the policer. There is only support for the policer levels 1 and 4.
<type>	Specifies the policer type.
<flag>	Specifies the policer flags.
level1-rates cir <cir>	Specifies the committed information rate (CIR) in Kilobits per second (Kbps) for level-1. The same is applicable for level-2 to level-4.
level1-rates pir <pir>	Specifies the peak information rate (PIR) in Kilobits per second (Kbps) for level-1. The same is applicable for level-2 to level-4.
level1-rates cbs <cbs>	Specifies the Committed burst size (CBS) in Kilobits per second (Kbps) for level-1. The same is applicable for level-2 to level-4.
level1-rates pbs <pbs>	Specifies the peak burst size (PBS) in Kilobits per second (Kbps) for level-1. The same is applicable for level-2 to level-4.
level1-rates max-cir <max-cir>	Specifies the maximum for the level-1 committed information rate (CIR) in kilobits per second (Kbps). The same is applicable for level-2 to level-4.
level1-rates max-pir <max-pir>	Specifies the maximum for the level-1 peak information rate (PIR) in Kilobits per second (Kbps). The same is applicable for level-2 to level-4.

The following example configures the QoS policer policer-residential with multi-level policers (levels=4), CIR, PIR rates, and burst sizes (CBS, PBS) for each level (level-1, level-2, level-3 and level-4). Also, two-rate-three-color policer type and color-blind as the default configuration. The four-level Policer configuration is as follows:

- Level-1(cir=2Mbps, pir=2.5Mbps, cbs=1000, pbs=1000)
- Level-2(cir=3Mbps, pir=3.5Mbps, cbs=1000, pbs=1000)
- Level-3(cir=4Mbps, pir=4.5Mbps, cbs=1000, pbs=1000)
- Level-4(cir=1Mbps, pir=1.5Mbps, cbs=1000, pbs=1000)

```
set forwarding-options class-of-service policer policer-residential
set forwarding-options class-of-service policer policer-residential level1-rates cir 2000
set forwarding-options class-of-service policer policer-residential level1-rates cbs 1000
set forwarding-options class-of-service policer policer-residential level1-rates pir 2500
set forwarding-options class-of-service policer policer-residential level1-rates pbs 1000
set forwarding-options class-of-service policer policer-residential level2-rates cir 3000
set forwarding-options class-of-service policer policer-residential level2-rates cbs 1000
set forwarding-options class-of-service policer policer-residential level2-rates pir 3500
set forwarding-options class-of-service policer policer-residential level2-rates pbs 1000
set forwarding-options class-of-service policer policer-residential level3-rates cir 4000
set forwarding-options class-of-service policer policer-residential level3-rates cbs 1000
set forwarding-options class-of-service policer policer-residential level3-rates pir 4500
set forwarding-options class-of-service policer policer-residential level3-rates pbs 1000
set forwarding-options class-of-service policer policer-residential level4-rates cir 1000
set forwarding-options class-of-service policer policer-residential level4-rates cbs 1000
set forwarding-options class-of-service policer policer-residential level4-rates pir 1500
set forwarding-options class-of-service policer policer-residential level4-rates pbs 1000
set forwarding-options class-of-service policer policer-residential levels 4
set forwarding-options class-of-service policer policer-residential type two-rate-three-color
commit
```

The following example shows the QoS policer level rates configuration:

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service policer policer-residential
{
  "rtbrick-config:policer": [
{
            "policer-name": "policer-residential",
            "level1-rates": {
              "cir": 2000,
              "cbs": 1000,
              "pir": 2500,
              "pbs": 1000
            },
            "level2-rates": {
              "cir": 3000,
              "cbs": 1000,
              "pir": 3500,
              "pbs": 1000
            },
            "level3-rates": {
              "cir": 4000,
              "cbs": 1000,
              "pir": 4500,
              "pbs": 1000
            },
            "level4-rates": {
              "cir": 1000,
              "cbs": 1000,
              "pir": 1500,
              "pbs": 1000
            },
            "levels": 4,
            "type": "two-rate-three-color"
          }
  ]
}
supervisor@rtbrick>LEAF01: cfg>
```

13.4.2. Class-Policer-Map Configuration

Use the following CLI syntax to configure the Class-Policer-Map:

set forwarding-options class-of-service class-policer-map <class-policer-map-name> <attribue> <value>

Attribute	Description
<class-policer-map-name>	Specifies the class policer map name,
class <class>	Specifies the class such as class-0, class-1, class-2, class-3, class-4, class-5, class-6, class-7
class <class> <policer-level>	Specifies the policer levels. The supported levels are: level-1, level-2, level-3, and level-4

Below is an example configuration of the class-policer-map, which sets up level-1 to level-4 policer levels.

- class-0 mapped to policer level-1
- class-1 mapped to policer level-2
- class-2 mapped to policer level-3
- class-3 mapped to policer level-4

```
set forwarding-options class-of-service class-policer-map policer-map-residential class class-0 policer-level
level-1
set forwarding-options class-of-service class-policer-map policer-map-residential class class-1 policer-level
level-2
set forwarding-options class-of-service class-policer-map policer-map-residential class class-2 policer-level
level-3
set forwarding-options class-of-service class-policer-map policer-map-residential class class-3 policer-level
level-4
commit
```

The following example shows the class-policer-map configuration:

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service class-policer-map policer-
map-l2tp
{
  "rtbrick-config:class-policer-map": [
{
            "class-policer-map-name": "policer-map-residential",
            "class": [
              {
                "class": "class-0",
                "policer-level": "level-1"
              },
              {
                "class": "class-1",
                "policer-level": "level-2"
              },
              {
                "class": "class-2",
                "policer-level": "level-3"
              },
              {
                "class": "class-3",
                "policer-level": "level-4"
              }
            ]
          }
  ]
}
supervisor@rtbrick>LEAF01: cfg>
```

13.4.3. Queue Configuration

Use the following CLI syntax to configure a queue:

set forwarding-options class-of-service queue <queue-name> <atribute> <value>

Attribute	Description
<queue-name>	Specifies the queue name
queue-size <queue-size>	Specifies the size of the queue in bytes
<shaper-name>	(Optional) Specifies the shaper that is associated with the queue
wred minimum-threshold <minimum-threshold>	Specifies the minimum average queue size to apply WRED in bytes
wred maximum-threshold <maximum-threshold>	Specifies the maximum average queue size to apply WRED in bytes
wred drop-probability <drop-probability>	WRED drop probability applied at the maximum threshold
header-compensation bytes <bytes>	Specifies the header compensation value
header-compensation decrement true	Specifies whether the header compensation value is to be decremented.

The following example configures four queues with different traffic streams named BE_SUBS, LD_SUBS, LL_SUBS and VO_SUBS with queue attributes queue-size, queue header-compensation & queue shaper for Best Effort(BE), Low-Delay(LD), Low-Loss(LL) and Voice(VO) Queues.

```
set forwarding-options class-of-service queue BE_SUBS
set forwarding-options class-of-service queue BE_SUBS queue-size 375000
set forwarding-options class-of-service queue BE_SUBS header-compensation bytes 22
set forwarding-options class-of-service queue BE_SUBS header-compensation bytes 22 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 bytes 22 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 bytes 22 decrement true
set forwarding-options class-of-service queue VO_SUBS
set forwarding-options class-of-service queue VO_SUBS queue-size 156250
set forwarding-options class-of-service queue VO_SUBS header-compensation bytes 22
set forwarding-options class-of-service queue VO_SUBS header-compensation bytes 22 decrement true
set forwarding-options class-of-service queue VO_SUBS shaper-name shaper_VO
commit
```

The following example shows the queue configuration:

```
supervisor@DT-STD-23-2402>bm14-tst.fsn.rtbrick.net: cfg> show config forwarding-options class-of-service
queue
{
  "rtbrick-config:queue": [
{
            "queue-name": "BE_SUBS",
            "queue-size": 375000,
            "header-compensation": {
              "bytes": 22,
              "decrement": "true"
            }
          },
          {
            "queue-name": "LD_SUBS",
            "queue-size": 625000,
            "header-compensation": {
              "bytes": 22,
              "decrement": "true"
            }
          },
          {
            "queue-name": "LL_SUBS",
            "queue-size": 625000,
            "header-compensation": {
              "bytes": 22,
              "decrement": "true"
            }
          },
          {
            "queue-name": "VO_SUBS",
            "queue-size": 156250,
            "shaper-name": "shaper_VO",
            "header-compensation": {
              "bytes": 22,
              "decrement": "true"
            }
          }
  ]
}
supervisor@rtbrick>LEAF01: cfg>
```

13.4.4. Queue-Group Configuration

Queue group size: 1 or 4 or 8

Use the following CLI syntax to configure a queue group:

set forwarding-options class-of-service queue-group <queue-group-name> <attribute> <value>

Attribute	Description
<queue-group-name>	User-defined name for the queue-group
queue-numbers <queue-numbers>	Specifies the number of queues in a Queue Group

The following examples configure the queue group with queue numbers 1 and 4.

```
set forwarding-options class-of-service queue-group subs-4queues queue-numbers 1
commit
```

The following example shows the queue group configuration:

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service queue-group
{
  "rtbrick-config:queue-group": [
    {
      "queue-group-name": "subs-4queues",
      "queue-numbers": 1
    }
  ]
}
supervisor@rtbrick>LEAF01: cfg>
```

```
set forwarding-options class-of-service queue-group subs-4queues queue-numbers 4
commit
```

The following example shows the queue group configuration:

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service queue-group
{
  "rtbrick-config:queue-group": [
    {
      "queue-group-name": "subs-4queues",
      "queue-numbers": 4
    }
  ]
}
```

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

Class-Queue-Map Configuration

Use the following CLI syntax to configure the class-queue-map:

set forwarding-options class-of-service class-queue-map <class-queue-map-name> <attribute> <value>

Attribute	Description
<class-queue-map-name>	Specifies the class queue map name
class <class>	Specifies the class such as class-0, class-1, class-2, class-3, class-4, class-5, class-6, class-7
class <class> queue-name <queue-name>	Specifies the queue name

The following example configures the class-queue-map for the specific classes and queues:

- class-0 mapped to queue BE_SUBS
- class-1 mapped to queue LD_SUBS
- class-2 mapped to queue LL_SUBS
- class-3 mapped to queue VO_SUBS

```
set forwarding-options class-of-service class-queue-map subs-4queues class-0 queue-name BE_SUBS
set forwarding-options class-of-service class-queue-map subs-4queues class-1 queue-name LD_SUBS
set forwarding-options class-of-service class-queue-map subs-4queues class-2 queue-name LL_SUBS
set forwarding-options class-of-service class-queue-map subs-4queues class-3 queue-name VO_SUBS
commit
```

The following example shows the class-queue-map configuration:

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service
class-queue-map
{
  "rtbrick-config:class-queue-map": [
{
            "class-queue-map-name": "subs-4queues",
            "class": [
              {
                "class-type": "class-0",
                "queue-name": "BE_SUBS"
              },
              {
                "class-type": "class-1",
                "queue-name": "LD_SUBS"
              },
              {
                "class-type": "class-2",
                "queue-name": "LL_SUBS"
              },
              {
                "class-type": "class-3",
                "queue-name": "VO_SUBS"
              }
            ]
          }
  ]
}
supervisor@rtbrick>LEAF01: cfg>
```

13.4.5. Scheduler Configuration

Use the following CLI syntax to configure a scheduler:

set forwarding-options class-of-service scheduler <attribue> <value>

Attribute	Description
<scheduler-name>	User-defined Scheduler Name
<scheduler-name> shaper-name <shaper-name>	(Optional) User-defined Shaper Name
<scheduler-name> type <type>	Specifies the Scheduler Type 2sp_wfq_discrete 3sp_wfq_discrete strict_priority wfq_discrete_2sp 2sp_wfq_independent fair_queueing weighted_fair_queueing wfq_independent_2sp
<scheduler-name> composite true	(Optional) keyword to specify the scheduler as composite type

The following example configures two schedulers: subs-4queues with scheduler type strict_priority for the Subscriber level, and pon0 with scheduler type fair_queueing for PON/GPON level.

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

The following example shows the QoS scheduler configuration:

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service scheduler
{
  "rtbrick-config:scheduler": [
          {
            "scheduler-name": "pon0",
            "type": "fair_queueing"
          },
          {
            "scheduler-name": "subs-4queues",
            "shaper-name": "shaper_session",
            "type": "strict_priority"
          }
  ]
}
supervisor@rtbrick>LEAF01: cfg>
```

Priority Propagation

Use the following CLI syntax to configure priority propagation:

set forwarding-options class-of-service global priority-propagation [**enable** | **disable**]

The following example enables priority propagation at the global level to operate schedulers in dual-flow mode, with high-priority and low-priority flows.

```
set forwarding-options class-of-service global priority-propagation enable
```

The following example shows the priority propagation:

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service global
{
  "rtbrick-config:global": {
    "priority-propagation": "enable"
  }
}
supervisor@rtbrick>LEAF01: cfg>
```

13.4.6. Shaper Configuration

Use the following CLI syntax to configure a shaper:

set forwarding-options class-of-service shaper <shaper-name> <attribue> <value>

Command Arguments

Attribute	Description
<shaper-name>	User-defined shaper name
<shaping-rate-high>	High flow shaping rate in kilobits per second
<shaping-rate-low>	Low flow shaping rate in kilobits per second

- If priority propagation is not enabled, high-flow shaping value will be considered for shaper.
- If the scheduler type is strict_priority, the mapping of queues to priorities begins with strict_priority_1.

The following example configures the shaper high-flow and low-flow rates for Subscriber Session Level with shaper-name shaper_session and Queue Level with shaper-name shaper_VO.

```
set forwarding-options class-of-service shaper shaper_session
set forwarding-options class-of-service shaper shaper_session shaping-rate-high 10000
set forwarding-options class-of-service shaper shaper_session shaping-rate-low 100
set forwarding-options class-of-service shaper shaper_VO
set forwarding-options class-of-service shaper shaper_VO shaping-rate-high 2000
set forwarding-options class-of-service shaper shaper_VO shaping-rate-low 0
commit
```

The following example shows the shaper configuration:

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service shaper
```

```
{
  "rtbrick-config:shaper": [
          {
            "shaper-name": "shaper_VO",
            "shaping-rate-high": 2000,
            "shaping-rate-low": 0
          },
          {
            "shaper-name": "shaper_session",
            "shaping-rate-high": 10000,
            "shaping-rate-low": 100
          }
  ]
}
supervisor@rtbrick>LEAF01: cfg>
```

13.4.7. Scheduler-Map Configuration

Use the following CLI syntax to configure a Scheduler-Map:

set forwarding-options class-of-service scheduler-map <scheduler-map-name> <attribue> <value>

Command arguments

Attribute	Description
<scheduler-map-name>	Specifies the name of the scheduler-map
scheduler-name <scheduler-name>	Specifies the name of the scheduler
queue-group-name <group-name>	Specifies the name of the queue-group
queue-group-name <group-name> queue-name <name>	Specifies the name of the queue

Attribute	Description
queue-group-name <group-name> queue-name <name> connection-point <connection-point>	Specifies the type of connection point, such as no_priority, strict_priority_0, strict_priority_1, strict_priority_2, strict_priority
queue-group-name <group-name> queue-name <name> parent-flow <high-flow / low-flow>	(Optional) Specifies the type of the parent flow, that is high-flow or low-flow.
queue-group-name <group-name> queue-name <name> parent-scheduler-name <parent-scheduler-name>	Specifies the name of the parent scheduler
queue-group-name <group-name> queue-name <name> port-connection	<port-connection-type>
Specifies the type of port connection, that is, queue_to_port or scheduler_to_port	queue-group-name <group-name> queue-name <name> weight <weight>

The following example configures a scheduler map for OLT with schedmap-olt and for subscribers with subs-4queues-residential. OLT scheduler-map schedmap-olt is directly connected to the physical port and subs-4queues-residential connects different queues (BE_SUBS, LL_SUBS,LD_SUBS,VO_SUBS) with different connection-points(strict_priority_3, strict_priority_2, strict_priority_1, strict_priority_0).

```
set forwarding-options class-of-service scheduler-map schedmap-olt
set forwarding-options class-of-service scheduler-map schedmap-olt scheduler-name pon0
set forwarding-options class-of-service scheduler-map schedmap-olt scheduler-name pon0 port-connection
scheduler_to_port
set forwarding-options class-of-service scheduler-map subs-4queues-residential
set forwarding-options class-of-service scheduler-map subs-4queues-residential queue-group-name subs-4queues
set forwarding-options class-of-service scheduler-map subs-4queues-residential queue-group-name subs-4queues
queue-name BE_SUBS
set forwarding-options class-of-service scheduler-map subs-4queues-residential queue-group-name subs-4queues
queue-name BE_SUBS parent-flow high-flow
set forwarding-options class-of-service scheduler-map subs-4queues-residential queue-group-name subs-4queues
queue-name BE_SUBS parent-scheduler-name subs-4queues
set forwarding-options class-of-service scheduler-map subs-4queues-residential queue-group-name subs-4queues
queue-name BE_SUBS connection-point strict_priority_3
set forwarding-options class-of-service scheduler-map subs-4queues-residential queue-group-name subs-4queues
queue-name LD_SUBS
set forwarding-options class-of-service scheduler-map subs-4queues-residential queue-group-name subs-4queues
queue-name LD_SUBS parent-flow high-flow
set forwarding-options class-of-service scheduler-map subs-4queues-residential queue-group-name subs-4queues
queue-name LD_SUBS parent-scheduler-name subs-4queues
set forwarding-options class-of-service scheduler-map subs-4queues-residential queue-group-name subs-4queues
queue-name LD_SUBS connection-point strict_priority_1
set forwarding-options class-of-service scheduler-map subs-4queues-residential queue-group-name subs-4queues
queue-name LL_SUBS
set forwarding-options class-of-service scheduler-map subs-4queues-residential queue-group-name subs-4queues
queue-name LL_SUBS parent-flow high-flow
set forwarding-options class-of-service scheduler-map subs-4queues-residential queue-group-name subs-4queues
queue-name LL_SUBS parent-scheduler-name subs-4queues
set forwarding-options class-of-service scheduler-map subs-4queues-residential queue-group-name subs-4queues
queue-name LL_SUBS connection-point strict_priority_2
set forwarding-options class-of-service scheduler-map subs-4queues-residential queue-group-name subs-4queues
queue-name VO_SUBS
set forwarding-options class-of-service scheduler-map subs-4queues-residential queue-group-name subs-4queues
queue-name VO_SUBS parent-flow high-flow
set forwarding-options class-of-service scheduler-map subs-4queues-residential queue-group-name subs-4queues
queue-name VO_SUBS parent-scheduler-name subs-4queues
set forwarding-options class-of-service scheduler-map subs-4queues-residential queue-group-name subs-4queues
queue-name VO_SUBS connection-point strict_priority_0
set forwarding-options class-of-service scheduler-map subs-4queues-residential scheduler-name subs-4queues
set forwarding-options class-of-service scheduler-map subs-4queues-residential scheduler-name subs-4queues
port-connection scheduler_to_port
commit
```

The following example shows the scheduler-map configuration:

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service scheduler-map
{
  "rtbrick-config:scheduler-map": [
{
            "scheduler-map-name": "schedmap-olt",
            "scheduler-name": [
              {
                "name": "pon0",
                "port-connection": "scheduler_to_port"
              }
            ]
          },
          {
            "scheduler-map-name": "subs-4queues-residential",
            "queue-group-name": [
              {
                "group-name": "subs-4queues",
                "queue-name": [
                  {
                    "name": "BE_SUBS",
                    "parent-flow": "high-flow",
                    "parent-scheduler-name": "subs-4queues",
                    "connection-point": "strict_priority_3"
                  },
                  {
                    "name": "LD_SUBS",
                    "parent-flow": "high-flow",
                    "parent-scheduler-name": "subs-4queues",
                    "connection-point": "strict_priority_1"
                  },
                  {
                    "name": "LL_SUBS",
                    "parent-flow": "high-flow",
                    "parent-scheduler-name": "subs-4queues",
                    "connection-point": "strict_priority_2"
                  },
                  {
                    "name": "VO_SUBS",
                    "parent-flow": "high-flow",
                    "parent-scheduler-name": "subs-4queues",
                    "connection-point": "strict_priority_0"
                  }
                ]
              }
            ],
            "scheduler-name": [
              {
                "name": "subs-4queues",
                "port-connection": "scheduler_to_port"
              }
            ]
          }
  ]
}
supervisor@rtbrick>LEAF01: cfg>
```

Remark-Map Configuration

Use the following CLI syntax to configure the remark-map:

set forwarding-options class-of-service remark-map <remark-map-name> <attribute> <value>

Command arguments

Attribute	Description
<remark-map-name>	Specifies the remaking map name
remark-type <remark-type>	Specifies the remarking type - ipv4-tos, ipv6-tc, mpls-ipv4, mpls-ipv6, ieee-802.1

Attribute	Description
remark-type <remark-type> <match-codepoint>	Specifies the match code point for the specified remarking type. On the UfiSpace S9600-72XC, UfiSpace S9600-32X, and Delta AGCVA48S platforms, the match codepoint is TOS for VLAN IEEE-802.1p remarking.
remark-type <remark-type> <match-codepoint> color <color>	Indicates the color - all, green, yellow. Color is used to set different remark codepoints for the same match-codepoint based on color marked by the Policer.
remark-type <remark-type> <match-codepoint> color <color> remark-codepoint <remark-codepoint>	Specifies the remarking codepoint

In the following example, the remark map subs-remarking-residential is configured with match-codepoint as 128, 160, 192, and 224. The color is set to "all", and the remark-codepoint is 6.

```
set forwarding-options class-of-service remark-map subs-remarking-residential
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-
codepoint 128
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-
codepoint 128 color all
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-
codepoint 128 color all remark-codepoint 6
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-
codepoint 160
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-
codepoint 160 color all
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-
codepoint 160 color all remark-codepoint 6
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-
codepoint 192
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-
codepoint 192 color all
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-
codepoint 192 color all remark-codepoint 6
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-
codepoint 224
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-
codepoint 224 color all
set forwarding-options class-of-service remark-map subs-remarking-residential remark-type ieee-802.1 match-
codepoint 224 color all remark-codepoint 6
commit
```

The following example shows the remark map configuration:

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service remark-map remark-exp remark-
type ipv6-tc
{
  "rtbrick-config:remark-type": [
{
            "remark-map-name": "subs-remarking-residential",
            "remark-type": [
              {
                "remark-type": "ieee-802.1",
                "match-codepoint": [
                  {
                    "match-codepoint": 128,
                    "color": [
                      {
                        "color": "all",
                        "remark-codepoint": 6
                      }
                    ]
                  },
                  {
                    "match-codepoint": 160,
                    "color": [
                      {
                        "color": "all",
                        "remark-codepoint": 6
                      }
                    ]
                  },
                  {
                    "match-codepoint": 192,
                    "color": [
                      {
                        "color": "all",
                        "remark-codepoint": 6
                      }
                    ]
                  },
                  {
                    "match-codepoint": 224,
                    "color": [
                      {
                        "color": "all",
                        "remark-codepoint": 6
                      }
                    ]
                  }
                ]
              }
            ]
          }
  ]
}
supervisor@rtbrick>LEAF01: cfg>
```

Global Profile Mapping

Use the following CLI syntax to configure the remark map for a global profile.

set forwarding-options class-of-service global remark-map-name <remark-map-name>

Command arguments

Attribute	Description
<remark-map-name>	Specifies the name of the remark map.

The following example configures the remark map for a global profile:

```
set forwarding-options class-of-service global remark-map-name subs-remarking-residential
commit
```

The following example shows the remark map for a global profile configuration:

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service global
{
  "rtbrick-config:global": {
    "multifield-classifier-name": "global_mfc",
    "remark-map-name": "subs-remarking-residential"
  }
}
supervisor@rtbrick>LEAF01: cfg>
```

Remark-map to Instance Mapping

Use the following CLI syntax to configure the remark map for an instance.

set forwarding-options class-of-service instance <instance-name> **remark-map-name** <remark-map-name>

Command arguments

Attribute	Description
<instance-name>	Specifies the name of the instance.
<remark-map-name>	Specifies the name of the remark map.

The following example configures the remark map for the instance default.

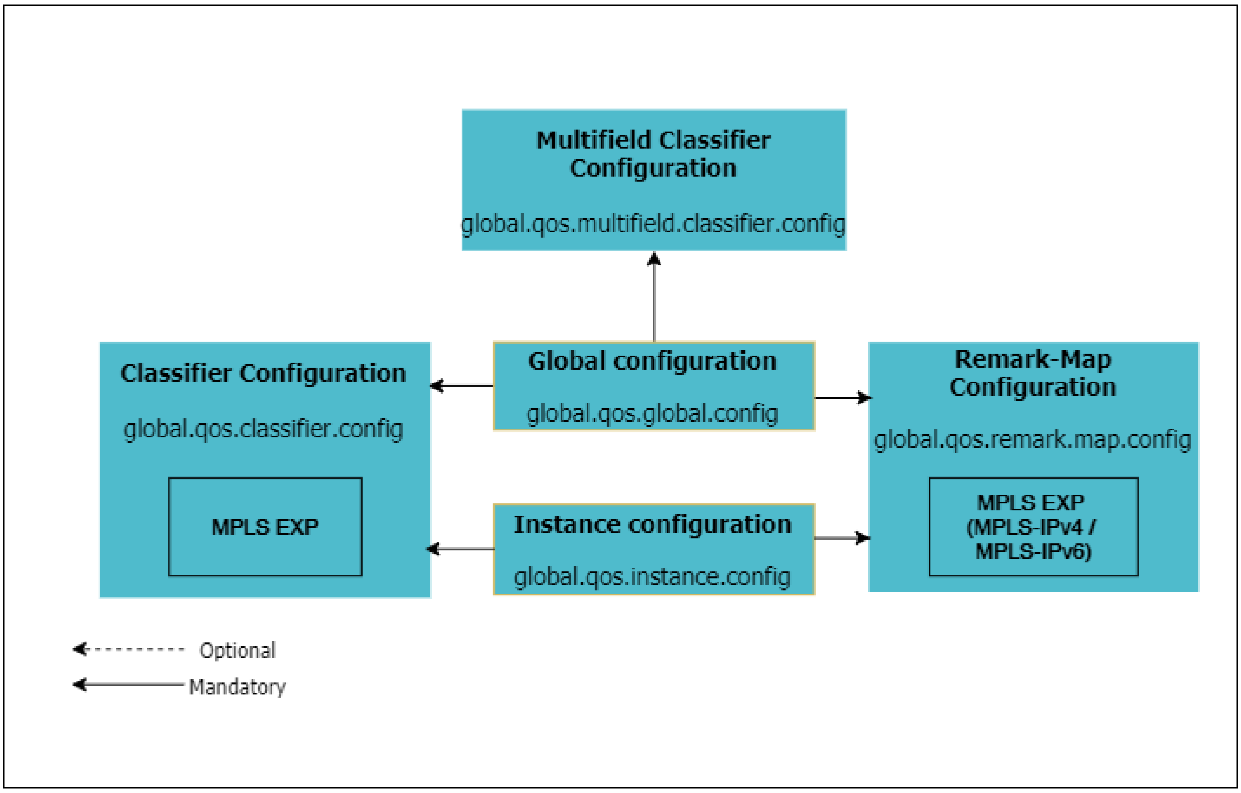
```
set forwarding-options class-of-service instance default remark-map-name subs-remarking-residential
commit
```

The following example shows the remark map configuration for the instance default.

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service instance default
{
  "rtbrick-config:instance": [
    {
      "name": "default",
      "remark-map-name": "subs-remarking-residential"
    }
  ]
}
supervisor@rtbrick>LEAF01: cfg>
```

QoS Global and Instance Configurations

The figure below shows the dependencies for per instance or global classifier and remark-map configurations.


Multifield Classifier Configuration
global.qos.multifield.classifier.config
Classifier Configuration
global.qos.classifier.config
MPLS EXP
Global configuration
global.qos.global.config
Instance configuration
global.qos.instance.config
Remark-Map Configuration
global.qos.remark.map.config
MPLS EXP (MPLS-IPv4 / MPLS-IPv6)
Optional
Mandatory

The following example configures the BA classifier subs-exp-class for traffic exp with match code point 2 classified as class class-1.

```
set forwarding-options class-of-service classifier subs-exp-class match-type exp
codepoint 2 class class-1
```

BA Classifier to Global Mapping

The MPLS classifiers can be applied globally using global configuration.

```
set forwarding-options class-of-service global classifier-name <classifier-name>
```

Attribute	Description
<classifier-name>	Specifies the classifier name

The following example shows how to configure the subs-exp-class BA Classifier for global mapping.

```
set forwarding-options class-of-service global classifier-name subs-exp-class
commit
```

The configuration for global mapping to the BA Classifier is shown in the following example.

```
supervisor@rtbrick>LEAF01: cfg> show config  forwarding-options class-of-service global classifier-name
{
  "rtbrick-config:classifier-name": "subs-exp-class"
}
supervisor@rtbrick>LEAF01: cfg>
```

Single Queue To Global Mapping

The Single-Queue can be enabled with global configuration.

For the system to initialize with a single queue in a Queue-Group, a system reboot is required.

```
set forwarding-options class-of-service global queue-group-profile single-queue
```

Attribute	Description

The following example shows how to enable single-queue in a Queue-Group.

```
set forwarding-options class-of-service global queue-group-profile single-queue
commit
```

The configuration for a single queue in Queue-Group is shown in the following example.

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service global
{
  "rtbrick-config:global": {
    "queue-group-profile": "single-queue"
  }
}
supervisor@rtbrick>LEAF01: cfg>
```

Remark-Map To Global Mapping

Use the following CLI syntax to configure the remark map for a global profile.

set forwarding-options class-of-service global remark-map-name <remark-map-name>

Command arguments

Attribute	Description
<remark-map-name>	Specifies the name of the remark map.

The following example configures the remark map for a global profile:

```
set forwarding-options class-of-service global remark-map-name subs-remarking-residential
commit
```

Multifield Classifier (MFC) to Global Mapping

Use the following CLI syntax to configure the MF Classifier to Global Mapping:

set forwarding-options class-of-service global multifield-classifier-name <multifield-classifier-name>

Attribute	Description
<multifield-classifier-name>	Specifies the name of the multifield classifier

The following example configures the multifield Classifier to Global Mapping:

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

The following example shows the multifield classifier to Global Mapping configuration:

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service global
{
  "rtbrick-config:global": {
    "multifield-classifier-name": "global_mfc"
  }
}
supervisor@rtbrick>LEAF01: cfg>
```

BA Classifier to Instance Mapping

The MPLS classifiers can be applied at an instance level using instance configuration.

```
set forwarding-options class-of-service instance <instance> classifier-name
<classifier-name>
```

Attribute	Description
<instance>	Specifies the instance name
<classifier-name>	Specifies the classifier name

The following example shows how to configure the subs-exp-class BA Classifier for the instance default.

```
set forwarding-options class-of-service instance default classifier-name subs-exp-class
commit
```

The configuration for instance mapping to the BA Classifier is shown in the following example.

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service instance default
{
  "rtbrick-config:instance": [
    {
      "name": "default",
      "classifier-name": "subs-exp-class"
    }
  ]
}
supervisor@rtbrick>LEAF01: cfg>
```

Remark-Map To Instance Mapping

Use the following CLI syntax to configure the remark map for an instance.

set forwarding-options class-of-service instance <instance-name> remark-map-name <remark-map-name>

Command arguments

Attribute	Description
<instance-name>	Specifies the name of the instance.
<remark-map-name>	Specifies the name of the remark map.

The following example configures the remark map for the instance default:

```
set forwarding-options class-of-service instance default remark-map-name subs-remarking-residential
commit
```

The configuration for the remark map for an instance is shown in the following example.

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service instance default
{
  "rtbrick-config:instance": [
    {
      "name": "default",
      "remark-map-name": "subs-remarking-residential"
    }
  ]
}
supervisor@rtbrick>LEAF01: cfg>
```

13.4.8. HQoS Show Running-Configuration

To display the running configuration, use the **show config** command.

Syntax

show config

Example

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options
    "rtbrick-config:forwarding-options": {
      "mirror": [
        {
          "name": "m1",
          "destination": {
            "interface": "cpu-0/0/200"
          },
          "source": {
            "direction": "ingress",
            "interface": "ifp-0/1/30"
          }
        }
      ],
      "class-of-service": {
        "classifier": [
          {
            "classifier-name": "subs-pbit-class",
            "match-type": [
              {
                "match-type": "ieee-802.1",
                "codepoint": [
                  {
                    "codepoint": 1,
                    "class": "class-0",
                    "remark-codepoint": 7
                  },
                  {
                    "codepoint": 2,
                    "class": "class-1",
                    "remark-codepoint": 7
                  },
                  {
                    "codepoint": 3,
                    "class": "class-2",
                    "remark-codepoint": 7
                  },
                  {
                    "codepoint": 4,
                    "class": "class-3",
                    "remark-codepoint": 7
                  }
                ]
              }
            ]
          }
        ],
        "class-policer-map": [
          {
            "class-policer-map-name": "policer-map-residential",
            "class": [
              {
                "class": "class-0",
                "policer-level": "level-1"
              },
              {
                "class": "class-1",
                "policer-level": "level-2"
              },
              {
                "class": "class-2",
                "policer-level": "level-3"
              },
              {
                "class": "class-3",
                "policer-level": "level-4"
              }
            ]
          }
        ],
        "class-queue-map": [
          {
            "class-queue-map-name": "subs-4queues",
            "class": [
              {
                "class-type": "class-0",
                "queue-name": "BE_SUBS"
              },
              {
                "class-type": "class-1",
                "queue-name": "LD_SUBS"
              },
              {
                "class-type": "class-2",
                "queue-name": "LL_SUBS"
              },
              {
                "class-type": "class-3",
                "queue-name": "VO_SUBS"
              }
            ]
          }
        ],
        "global": {
          "multifield-classifier-name": "global_mfc"
        },
        "policer": [
          {
            "policer-name": "policer-residential",
            "level1-rates": {
              "cir": 2000,
              "cbs": 1000,
              "pir": 2500,
              "pbs": 1000
            },
            "level2-rates": {
              "cir": 3000,
              "cbs": 1000,
              "pir": 3500,
              "pbs": 1000
            },
            "level3-rates": {
              "cir": 4000,
              "cbs": 1000,
              "pir": 4500,
              "pbs": 1000
            },
            "level4-rates": {
              "cir": 1000,
              "cbs": 1000,
              "pir": 1500,
              "pbs": 1000
            },
            "levels": 4,
            "type": "two-rate-three-color"
          }
        ],
        "profile": [
          {
            "profile-name": "residential",
            "classifier-name": "subs-pbit-class",
            "class-queue-map-name": "subs-4queues",
            "remark-map-name": "subs-remarking-residential",
            "class-policer-map-name": "policer-map-residential",
            "policer-name": "policer-residential",
            "scheduler-map-name": "subs-4queues-residential"
          }
        ],
        "queue": [
          {
            "queue-name": "BE_SUBS",
            "queue-size": 375000
          },
          {
            "queue-name": "LD_SUBS",
            "queue-size": 625000
          },
          {
            "queue-name": "LL_SUBS",
            "queue-size": 625000
          },
          {
            "queue-name": "VO_SUBS",
            "queue-size": 156250,
            "shaper-name": "shaper_VO"
          }
        ],
        "queue-group": [
          {
            "queue-group-name": "subs-4queues",
            "queue-numbers": 4
          }
        ],
        "remark-map": [
          {
            "remark-map-name": "subs-remarking-residential",
            "remark-type": [
              {
                "remark-type": "ieee-802.1",
                "match-codepoint": [
                  {
                    "match-codepoint": 128,
                    "color": [
                      {
                        "color": "all",
                        "remark-codepoint": 6
                      }
                    ]
                  },
                  {
                    "match-codepoint": 160,
                    "color": [
                      {
                        "color": "all",
                        "remark-codepoint": 6
                      }
                    ]
                  },
                  {
                    "match-codepoint": 192,
                    "color": [
                      {
                        "color": "all",
                        "remark-codepoint": 6
                      }
                    ]
                  },
                  {
                    "match-codepoint": 224,
                    "color": [
                      {
                        "color": "all",
                        "remark-codepoint": 6
                      }
                    ]
                  }
                ]
              }
            ]
          }
        ],
        "scheduler": [
          {
            "scheduler-name": "pon0",
            "shaper-name": "gpon-shaper",
            "type": "fair_queueing"
          },
          {
            "scheduler-name": "subs-4queues",
            "shaper-name": "shaper_session",
            "type": "strict_priority",
            "composite": "false"
          }
        ],
        "scheduler-map": [
          {
            "scheduler-map-name": "schedmap-olt",
            "scheduler-name": [
              {
                "name": "pon0",
                "port-connection": "scheduler_to_port"
              }
            ]
          },
          {
            "scheduler-map-name": "subs-4queues-residential",
            "queue-group-name": [
              {
                "group-name": "subs-4queues",
                "queue-name": [
                  {
                    "name": "BE_SUBS",
                    "parent-flow": "high-flow",
                    "parent-scheduler-name": "subs-4queues",
                    "connection-point": "strict_priority_3"
                  },
                  {
                    "name": "LD_SUBS",
                    "parent-flow": "high-flow",
                    "parent-scheduler-name": "subs-4queues",
                    "connection-point": "strict_priority_1"
                  },
                  {
                    "name": "LL_SUBS",
                    "parent-flow": "high-flow",
                    "parent-scheduler-name": "subs-4queues",
                    "connection-point": "strict_priority_2"
                  },
                  {
                    "name": "VO_SUBS",
                    "parent-flow": "high-flow",
                    "parent-scheduler-name": "subs-4queues",
                    "connection-point": "strict_priority_0"
                  }
                ]
              }
            ],
            "scheduler-name": [
              {
                "name": "subs-4queues",
                "port-connection": "scheduler_to_port"
              }
            ]
          }
        ],
        "shaper": [
          {
            "shaper-name": "shaper_VO",
            "shaping-rate-high": 2000,
            "shaping-rate-low": 0
          },
          {
            "shaper-name": "shaper_session",
            "shaping-rate-high": 10000,
            "shaping-rate-low": 100
          },
          {
            "shaper-name": "gpon-shaper",
            "shaping-rate-high": 2488000,
            "shaping-rate-low": 32000
          }
        ],
        "multifield-classifier": {
          "acl": {
            "l3v4": {
              "rule": [
                {
                  "rule-name": "global_mfc",
                  "ordinal": [
                    {
                      "ordinal-value": 1001,
                      "match": {
                        "ipv4-tos": 128,
                        "source-ipv4-prefix": "192.0.2.2/32"
                      },
                      "action": {
                        "forward-class": "class-0"
                      }
                    },
                    {
                      "ordinal-value": 1002,
                      "match": {
                        "ipv4-tos": 160,
                        "source-ipv4-prefix": "192.0.2.2/32"
                      },
                      "action": {
                        "forward-class": "class-1"
                      }
                    },
                    {
                      "ordinal-value": 1003,
                      "match": {
                        "ipv4-tos": 192,
                        "source-ipv4-prefix": "192.0.2.2/32"
                      },
                      "action": {
                        "forward-class": "class-2"
                      }
                    },
                    {
                      "ordinal-value": 1004,
                      "match": {
                        "ipv4-tos": 224,
                        "source-ipv4-prefix": "192.0.2.2/32"
                      },
                      "action": {
                        "forward-class": "class-3"
                      }
                    }
                  ]
                }
              ]
            }
          }
        }
      }
    }
```

13.4.9. HQoS Operational Commands

HQoS Show Commands

The HQoS show commands provide detailed information about the HQoS operation.

show qos classifier

Syntax:

show qos classifier <option>

Option	Description
-	Without any option, this command displays a summary of all the classifiers.
<classifier-name>	Displays QoS classifier information for the specified classifier.

The following example displays a summary of all the HQoS classifiers.

```
supervisor@rtbrick>LEAF01: op> show qos classifier
Classifier: residential-ip-classifier
Active: False
  Match Type   Codepoint    Class      Remark Codepoint    Color
  ipv4-tos     0            class-0    -                   -
  ipv4-tos     32           class-1    -                   -
  ipv4-tos     64           class-2    -                   -
  ipv4-tos     96           class-3    -                   -
  ipv4-tos     128          class-4    -                   -
  ipv4-tos     160          class-5    -                   -
  ipv4-tos     192          class-6    -                   -
  ipv4-tos     224          class-7    -                   -
  ipv6-tc      0            class-0    -                   -
  ipv6-tc      32           class-1    -                   -
  ipv6-tc      64           class-2    -                   -
  ipv6-tc      96           class-3    -                   -
  ipv6-tc      128          class-4    -                   -
  ipv6-tc      160          class-5    -                   -
  ipv6-tc      192          class-6    -                   -
  ipv6-tc      224          class-7    -                   -
Classifier: residential-pbit-classifier
Active: True
  Match Type   Codepoint    Class      Remark Codepoint    Color
  ieee-802.1   0            class-0    -                   -
  ieee-802.1   1            class-1    -                   -
  ieee-802.1   2            class-2    -                   -
  ieee-802.1   3            class-3    -                   -
  ieee-802.1   4            class-4    -                   -
  ieee-802.1   5            class-5    -                   -
  ieee-802.1   6            class-6    -                   -
  ieee-802.1   7            class-7    -                   -
```

The following example displays information for the specified classifier.

```
supervisor@rtbrick>LEAF01: op> show qos classifier residential-pbit-classifier
Classifier: residential-pbit-classifier
Active: True
  Match Type   Codepoint    Class      Remark Codepoint    Color
  ieee-802.1   0            class-0    -                   -
  ieee-802.1   1            class-1    -                   -
  ieee-802.1   2            class-2    -                   -
  ieee-802.1   3            class-3    -                   -
  ieee-802.1   4            class-4    -                   -
  ieee-802.1   5            class-5    -                   -
  ieee-802.1   6            class-6    -                   -
  ieee-802.1   7            class-7    -                   -
```

show qos interface

Syntax:

show qos interface <option>

Option	Description
-	Without any option, this command displays a summary of all the interfaces.
<interface-name>	Displays QoS classifier information for the specified interface.

The following example displays information for the specified interface.

```
supervisor@rtbrick>LEAF01: op> show qos interface ifl-0/0/10/200
Interface           Profile
ifl-0/0/10/200      pta_8queues_comp_on_S
supervisor@rtbrick>LEAF01: op>
```

show qos policer

Syntax:

show qos policer <option>

Option	Description
-	Without any option, this command displays a summary of all the QoS policers.
<policer-name>	Displays QoS policer information.
counter	Displays policer counter information.
class-to-policer-map	Displays class-to-policer-map information.

The following example displays a summary of all the QoS policers.

```
supervisor@rtbrick>LEAF01: op> show qos policer
Policer: _DEFAULT_POLICER_50_MB
Active: True, Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(KB)        PBS(KB)        Max CIR(Kbps)  Max PIR(Kbps)
  1        50000          50000          33000          33000          -              -
  2        -              -              -              -              -              -
  3        -              -              -              -              -              -
  4        -              -              -              -              -              -
Policer: policer-residential
Active: True, Type: two-rate-three-color, Levels: 4, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(KB)        PBS(KB)        Max CIR(Kbps)  Max PIR(Kbps)
  1        8000           8000           800            800            -              -
  2        -              -              -              -              -              -
  3        -              -              800            800            -              -
  4        -              -              800            800            -              -
```

The following example displays policer information for the specified policer.

```
supervisor@rtbrick>LEAF01: op> show qos policer policer-residential
Policer: policer-residential
Active: True, Type: two-rate-three-color, Levels: 4, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(KB)        PBS(KB)        Max CIR(Kbps)  Max PIR(Kbps)
  1        8000           8000           800            800            -              -
  2        -              -              -              -              -              -
  3        -              -              800            800            -              -
  4        -              -              800            800            -              -
supervisor@rtbrick>LEAF01: op>
```

The following example displays counter information for the specified counter.

```
supervisor@rtbrick>LEAF01: op> show qos policer counter lag-1
Interface                     Level  Units     Total         Received      Dropped
lag-1                         1      Packets   15773         15773         0
                                     Bytes     2555226       2555226       0
lag-1                         2      Packets   15778         15778         0
                                     Bytes     2556036       2556036       0
lag-1                         3      Packets   15775         15775         0
                                     Bytes     2555550       2555550       0
lag-1                         4      Packets   23661         23661         0
                                     Bytes     3423166       3423166       0
lag-1-egress                  1      Packets   7889          7889          0
                                     Bytes     1420020       1420020       0
lag-1-egress                  2      Packets   7889          7889          0
                                     Bytes     1420020       1420020       0
```

```
lag-1-egress                       3     Packets    7889              7889             0
                                         Bytes      1420020           1420020          0
lag-1-egress                       4     Packets    49177             49177            0
                                         Bytes      8288974           8288974          0
```

show qos profile

Syntax:

show qos profile <option>

Option	**Description**
-	Without any option, this command displays a summary of all the QoS profiles.
<profile-name>	Displays QoS profile information.

The following example displays a summary of all the QoS profiles.

```
supervisor@rtbrick>LEAF01: op> show qos profile lac_4queues_4classes
Profile: lac_4queues_4classes
   Classifier: residential-pbit-classifier
   Policer: policer-residential
   Scheduler map: lac_4queues_M
   Class queue map: lac_4queues_M
   Remark map: -
   Class policer map: policer-map-l2tp
   Mulifield classifier: -
supervisor@rtbrick>LEAF01: op>
```

show qos queue

Syntax:

show qos queue <option>

Option	**Description**
-	Without any option, this command displays a summary of all the queues.
class-to-queue-map	Displays queue class-to-queue-map information.
counter	Displays QoS queue counter information.
counter <interface-name>	Displays QoS queue counter information for the specified interface.

Option	Description
<interface-name>	Displays QoS queue details for the specified interface.

The following example displays a summary of all the queues.

```
supervisor@rtbrick>LEAF01: op> show qos queue
Applied queues:
  Interface          Queue              Queue Size         Min Thres          Max Thres          Drop Prob
Shaper
  ifl-0/0/10/100     BE_S               240000             -                  -                  -
-
  ifl-0/0/10/100     LD_S               200000             -                  -                  -
shaper_LD
  ifl-0/0/10/100     LL_S               200000             -                  -                  -
shaper_LL
  ifl-0/0/10/100     VO_S               50000              -                  -                  -
shaper_VO
  ifl-0/0/10/200     BE_S               240000             -                  -                  -
-
  ifl-0/0/10/200     LD_S               200000             -                  -                  -
shaper_LD
  ifl-0/0/10/200     LL_S               200000             -                  -                  -
shaper_LL
  ifl-0/0/10/200     VO_S               50000              -                  -                  -
shaper_VO
  ifl-0/0/10/300     BE_S               240000             -                  -                  -
-
  ifl-0/0/10/300     LD_S               200000             -                  -                  -
shaper_LD
  ifl-0/0/10/300     LL_S               200000             -                  -                  -
shaper_LL
  ifl-0/0/10/300     VO_S               50000              -                  -                  -
shaper_VO
Configured queues:
  Queue                Queue Size           Min Thres            Max Thres            Drop Prob
Shaper
  BE_L                 375000               -                    -                    -                    -
  BE_M                 375000               -                    -                    -                    -
  BE_S                 240000               -                    -                    -                    -
  CO_L                 312500               -                    -                    -                    -
  CO_M                 156250               -                    -                    -                    -
  CO_S                 50000                -                    -                    -                    -
  IO_L                 312500               -                    -                    -
shaper_IO
  IO_M                 156250               -                    -                    -
shaper_IO
  IO_S                 50000                -                    -                    -
shaper_IO
  LD_L                 1250000              -                    -                    -
shaper_LD
  LD_M                 625000               -                    -                    -
shaper_LD
  LD_S                 200000               -                    -                    -
shaper_LD
  LL_L                 1250000              -                    -                    -
shaper_LL
  LL_M                 625000               -                    -                    -
shaper_LL
  LL_S                 200000               -                    -                    -
shaper_LL
  VO_L                 312500               -                    -                    -
shaper_VO
  VO_M                 156250               -                    -                    -
shaper_VO
  VO_S                 50000                -                    -                    -
shaper_VO
  free_6_L             375000               -                    -                    -                    -
  free_6_M             375000               -                    -                    -                    -
  free_6_S             240000               -                    -                    -                    -
  free_7_L             375000               -                    -                    -                    -
  free_7_M             375000               -                    -                    -                    -
  free_7_S             240000               -                    -                    -                    -
```

The following example displays queue information for the specified interface.

```
supervisor@rtbrick>LEAF01: op> show qos queue ifl-0/0/10/100
Applied queues:
  Interface          Queue              Queue Size         Min Thres          Max Thres          Drop Prob
Shaper
  ifl-0/0/10/100     BE_S               240000             -                  -                  -
-
  ifl-0/0/10/100     LD_S               200000             -                  -                  -
shaper_LD
  ifl-0/0/10/100     LL_S               200000             -                  -                  -
shaper_LL
  ifl-0/0/10/100     VO_S               50000              -                  -                  -
shaper_VO
```

The following example displays queue counter information.

```
supervisor@rtbrick>LEAF01: op> show qos queue counter
Interface                      Queue Group            Queue                  Class    Units      Received
Queued              Dropped

                                                                                      Bytes      0
0                   0
ifl-0/1/32/6                   olt-dpu-mgmt-queues    OLT_MGMT               0        Packets    0
0                   0
                                                                                      Bytes      0
0                   0
ifl-0/1/33/4                   olt-dpu-mgmt-queues    OLT_MGMT               0        Packets    0
0                   0
                                                                                      Bytes      0
0                   0
ifl-0/1/33/6                   olt-dpu-mgmt-queues    OLT_MGMT               0        Packets    0
0                   0
                                                                                      Bytes      0
0                   0
ppp-0/1/30/72339069014638594   pta-4queues            BE_PTA                 0        Packets    941111
288270              652841
                                                                                      Bytes      942823712
288837428           653986284
ppp-0/1/30/72339069014638594   pta-4queues            LD_PTA                 2        Packets    938859
446474              492385
                                                                                      Bytes      942614436
448259896           494354540
ppp-0/1/30/72339069014638594   pta-4queues            LL_PTA                 1        Packets    938851
480506              458345
                                                                                      Bytes      942606404
482428024           460178380
ppp-0/1/30/72339069014638594   pta-4queues            VO_PTA                 3        Packets    3667257
673116              2994141
                                                                                      Bytes      953486820
175010160           778476660
l2bsa-0/1/30/281479271677953   l2bsa-4queues          BE_L2BSA               0        Packets    0
0                   0
                                                                                      Bytes      0
0                   0
l2bsa-0/1/30/281479271677953   l2bsa-4queues          LD_L2BSA               2        Packets    0
0                   0
                                                                                      Bytes      0
0                   0
l2bsa-0/1/30/281479271677953   l2bsa-4queues          LL_L2BSA               1        Packets    0
0                   0
                                                                                      Bytes      0
0                   0
```

The following example displays queue counter information for the specified interface.

```
supervisor@rtbrick>LEAF01: op> show qos queue counter ifl-0/1/30/6
Interface                      Queue Group            Queue                  Class    Units      Received
Queued              Dropped
ifl-0/1/30/6                   olt-dpu-mgmt-queues    OLT_MGMT               0        Packets    0
0                   0
                                                                                      Bytes      0
0                   0
```

show qos scheduler

Syntax:

show qos scheduler <option>

Option	Description
-	Without any option, this command displays a summary of all the schedulers.
<scheduler-name>	Displays scheduler information for the specified scheduler.

The following example displays a summary of all the schedulers.

```
supervisor@rtbrick>LEAF01: op> show qos scheduler
Scheduler              Type                   Shaper                 Composite            Active
fff                    strict_priority        -                      False                  False
fffd                   strict_priority        -                      False                  False
lac_4queues            strict_priority        -                      True                   False
olt-pon1               fair_queueing          -                      False                  False
olt-pon10              fair_queueing          -                      False                  False
olt-pon11              fair_queueing          -                      False                  False
olt-pon12              fair_queueing          -                      False                  False
olt-pon13              fair_queueing          -                      False                  False
olt-pon14              fair_queueing          -                      False                  False
olt-pon15              fair_queueing          -                      False                  False
olt-pon16              fair_queueing          -                      False                  False
olt-pon17              fair_queueing          -                      False                  False
olt-pon18              fair_queueing          -                      False                  False
olt-pon19              fair_queueing          -                      False                  False
olt-pon2               fair_queueing          -                      False                  False
olt-pon20              fair_queueing          -                      False                  False
olt-pon21              fair_queueing          -                      False                  False
olt-pon22              fair_queueing          -                      False                  False
olt-pon23              fair_queueing          -                      False                  False
olt-pon24              fair_queueing          -                      False                  False
olt-pon25              fair_queueing          -                      False                  False
olt-pon26              fair_queueing          -                      False                  False
olt-pon27              fair_queueing          -                      False                  False
olt-pon28              fair_queueing          -                      False                  False
olt-pon29              fair_queueing          -                      False                  False
olt-pon3               fair_queueing          -                      False                  False
olt-pon30              fair_queueing          -                      False                  False
olt-pon31              fair_queueing          -                      False                  False
```

```
olt-pon32              fair_queueing          -                      False                  False
olt-pon4               fair_queueing          -                      False                  False
olt-pon5               fair_queueing          -                      False                  False
olt-pon6               fair_queueing          -                      False                  False
olt-pon7               fair_queueing          -                      False                  False
olt-pon8               fair_queueing          -                      False                  False
olt-pon9               fair_queueing          -                      False                  False
pta_4queues_comp_off   strict_priority        -                      True                   False
pta_4queues_comp_on    strict_priority        -                      True                   False
pta_8queues_comp_off   strict_priority        -                      True                   False
pta_8queues_comp_on    strict_priority        -                      True                   False
supervisor@rtbrick>LEAF01: op>
```

The following example displays scheduler information for the specified scheduler.

```
supervisor@rtbrick>LEAF01: op> show qos scheduler lac_4queues
Scheduler              Type                   Shaper                 Composite            Active
lac_4queues            strict_priority        -                      True                   False
```

show qos scheduler-map

Syntax:

show qos scheduler-map <option>

Option	Description
-	Without any option, this command displays a summary of all the scheduler maps.
<scheduler-map>	Displays scheduler information for the specified scheduler map.

The following example displays a summary of all the scheduler maps.

```
supervisor@rtbrick>LEAF01: op> show qos scheduler-map
Scheduler-Map: lac_4queues_S
   Scheduler: fff                    Scheduler: strict_priority
     Queue: LD_S                     strict_priority_1
     Scheduler: pta_4queues_comp_off  Scheduler: strict_priority
       Queue: LL_S                     strict_priority_1
       Queue: VO_S                     strict_priority_0
   Scheduler: fffd                   Scheduler: strict_priority
     Queue: BE_S                     strict_priority_0
Scheduler-Map: schedmap-olt
   Scheduler: olt-pon1               Scheduler: fair_queueing
   Scheduler: olt-pon2               Scheduler: fair_queueing
   Scheduler: olt-pon3               Scheduler: fair_queueing
   Scheduler: olt-pon4               Scheduler: fair_queueing
   Scheduler: olt-pon5               Scheduler: fair_queueing
   Scheduler: olt-pon6               Scheduler: fair_queueing
   Scheduler: olt-pon7               Scheduler: fair_queueing
   Scheduler: olt-pon8               Scheduler: fair_queueing
   Scheduler: olt-pon9              Scheduler: fair_queueing
   Scheduler: olt-pon10             Scheduler: fair_queueing
   Scheduler: olt-pon11             Scheduler: fair_queueing
   Scheduler: olt-pon12             Scheduler: fair_queueing
   Scheduler: olt-pon13             Scheduler: fair_queueing
   Scheduler: olt-pon14             Scheduler: fair_queueing
   Scheduler: olt-pon15             Scheduler: fair_queueing
   Scheduler: olt-pon16             Scheduler: fair_queueing
   Scheduler: olt-pon17             Scheduler: fair_queueing
   Scheduler: olt-pon18             Scheduler: fair_queueing
   Scheduler: olt-pon19             Scheduler: fair_queueing
   Scheduler: olt-pon20             Scheduler: fair_queueing
   Scheduler: olt-pon21             Scheduler: fair_queueing
   Scheduler: olt-pon22             Scheduler: fair_queueing
   Scheduler: olt-pon23             Scheduler: fair_queueing
   Scheduler: olt-pon24             Scheduler: fair_queueing
   Scheduler: olt-pon25             Scheduler: fair_queueing
   Scheduler: olt-pon26             Scheduler: fair_queueing
   Scheduler: olt-pon27             Scheduler: fair_queueing
   Scheduler: olt-pon28             Scheduler: fair_queueing
   Scheduler: olt-pon29             Scheduler: fair_queueing
   Scheduler: olt-pon30             Scheduler: fair_queueing
   Scheduler: olt-pon31             Scheduler: fair_queueing
   Scheduler: olt-pon32             Scheduler: fair_queueing
   Scheduler: olt-pon33             Scheduler: False
Scheduler-Map: lac_4queues_L
   Scheduler: lac_4queues           Scheduler: strict_priority
     Queue: BE_L                    strict_priority_1
     Queue: LD_L                    strict_priority_1
     Queue: LL_L                    strict_priority_2
     Queue: VO_L                    strict_priority_0
```

The following example displays scheduler information for the specified scheduler-map.

```
supervisor@rtbrick>LEAF01: op> show qos scheduler-map lac_4queues_S
Scheduler-Map: lac_4queues_S
   Scheduler: fff                   Scheduler: strict_priority
     Queue: LD_S                    strict_priority_1
     Scheduler: pta_4queues_comp_off  Scheduler: strict_priority
       Queue: LL_S                    strict_priority_1
       Queue: VO_S                    strict_priority_0
   Scheduler: fffd                  Scheduler: strict_priority
     Queue: BE_S                    strict_priority_0
```

show qos shaper

Syntax:

show qos shaper <option>

Option	Description
-	Without any option, this command displays a summary of all the shapers.
<shaper-name>	Displays scheduler information for the specified shaper.

The following example displays a summary of all the shapers.

```
supervisor@rtbrick>LEAF01: op> show qos shaper
Shaper              High Rate(Kbps)     Low Rate(Kbps)      High Burst(Kb)      Low Burst(Kb)       Active
pon-shaper          2488000             -                   -                   -                   True
shaper_IO           -                   1000000             -                   -                   True
shaper_LD           1000000             -                   -                   -                   True
shaper_LL           1000000             -                   -                   -                   True
shaper_VO           1000000             -                   -                   -                   True
shaper_session      1000000             100                 -                   -                   True
```

The following example displays shaper information for the specified shaper.

```
supervisor@rtbrick>LEAF01: op> show qos shaper shaper_session
Shaper              High Rate(Kbps)     Low Rate(Kbps)      High Burst(Kb)      Low Burst(Kb)       Active
shaper_session      1000000             100                 -                   -                   False
```

show qos multifield-classifier

Syntax:

show qos multifield-classifier <option>

Option	Description
-	Without any option, this command displays a summary of all the multifield classifiers.
<multifield-classifier-name>	Displays scheduler information for the specified multifield classifiers.

The following example displays a summary of all the multifield classifiers.

```
supervisor@rtbrick>LEAF01: op> show qos multifield-classifier
Multifield Classifier: global-mfc
  ACL type: multifield_ipv4
  Ordinal: 10000
    Match:
      Source IPv4 prefix: 198.51.100.2/24
      IPv4 TOS: 224
    Action:
      Forward class: class-7
  Priority: 1000
  Ordinal: 9000
    Match:
      Source IPv4 prefix: 198.51.100.2/24
      IPv4 TOS: 192
    Action:
      Remark codepoint: 184
      Forward class: class-6
  Ordinal: 6200
    Match:
      Source IPv4 prefix: 198.51.100.2/24
      IPv4 TOS: 160
    Action:
      Remark codepoint: 184
      Forward class: class-5
  Ordinal: 5200
    Match:
      Source IPv4 prefix: 198.51.100.2/24
      IPv4 TOS: 64
    Action:
      Remark codepoint: 184
      Forward class: class-2
  Ordinal: 6000
    Match:
      Source IPv4 prefix: 198.51.100.2/24
      IPv4 TOS: 96
    Action:
      Remark codepoint: 184
      Forward class: class-3
  Ordinal: 5000
    Match:
      Source IPv4 prefix: 198.51.100.2/24
    Action:
      Remark codepoint: 184
      Forward class: class-0
  Ordinal: 6100
    Match:
      Source IPv4 prefix: 198.51.100.2/24
      IPv4 TOS: 128
    Action:
      Remark codepoint: 184
      Forward class: class-4
  Ordinal: 5100
    Match:
      Source IPv4 prefix: 198.51.100.2/24
      IPv4 TOS: 32
    Action:
      Remark codepoint: 184
      Forward class: class-1
  Ordinal: 100
    Match:
      Source IPv6 prefix: 2001:db8:0:100::/32
      IPv6 TC: 224
    Action:
      Remark codepoint: 196
      Forward class: class-3
Multifield Classifier: ipoe-double-play
  ACL type: multifield_ipv4
  Ordinal: 5200
    Match:
      Destination IPv4 prefix: 198.51.100.2/24
      IPv4 TOS: 160
    Action:
      Remark codepoint: 224
      Forward class: class-2
  Ordinal: 5000
    Match:
      Destination IPv4 prefix: 198.51.100.2/24
      IPv4 TOS: 64
    Action:
      Remark codepoint: 224
      Forward class: class-0
  Ordinal: 5300
    Match:
      Destination IPv4 prefix: 198.51.100.2/24
      IPv4 TOS: 192
    Action:
      Remark codepoint: 224
      Forward class: class-3
  Ordinal: 5100
    Match:
      Destination IPv4 prefix: 198.51.100.2/24
      IPv4 TOS: 96
    Action:
      Remark codepoint: 224
      Forward class: class-1
supervisor@rtbrick>LEAF01: op>
```

The following example displays multifield-classifier information for the specified multifield-classifier.

```
supervisor@rtbrick>LEAF01: op> show qos multifield-classifier global-mfc
Multifield Classifier: global-mfc
  ACL type: multifield_ipv4
  Ordinal: 10000
    Match:
      Source IPv4 prefix: 198.51.100.2/24
      IPv4 TOS: 224
    Action:
      Forward class: class-7
  Priority: 1000
  Ordinal: 9000
    Match:
      Source IPv4 prefix: 198.51.100.2/24
      IPv4 TOS: 192
    Action:
      Remark codepoint: 184
      Forward class: class-6
  Ordinal: 6200
    Match:
      Source IPv4 prefix: 198.51.100.2/24
      IPv4 TOS: 160
    Action:
      Remark codepoint: 184
      Forward class: class-5
  Ordinal: 5200
    Match:
      Source IPv4 prefix: 198.51.100.2/24
      IPv4 TOS: 64
    Action:
      Remark codepoint: 184
      Forward class: class-2
supervisor@rtbrick>LEAF01: op>
```

13.5. LLDP

13.5.1. LLDP Overview

Link Layer Discovery Protocol (LLDP) is a media-independent link layer protocol used by network devices for advertising their identity, capabilities to neighbors on a LAN segment. LLDP runs over the data-link layer only, allowing two systems running different network layer protocols to learn about each other.

LLDP supports a set of attributes that it uses to discover neighbor devices. These attributes contain type, length, and value descriptions and are referred to as TLVs. LLDP supported devices can use TLVs to receive and send information to their neighbors. Details such as configuration information, device capabilities, and device identity can be advertised using this protocol.

Guidelines

- All LLDP packets are sent to the CPU for further processing.
- LLDP packets will be lower priority packets when compared with the routing protocol packets. In case of a congestion, the LLDP packets may be policed.

Limitations

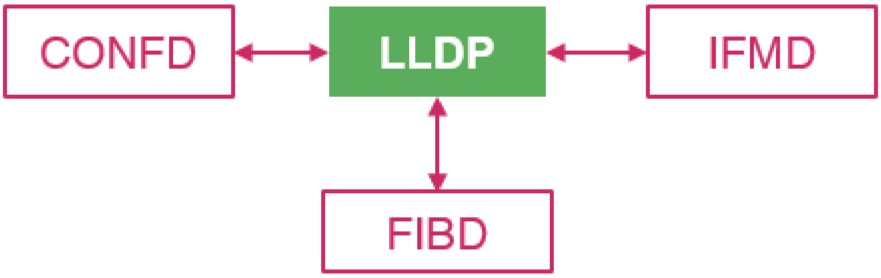
- LLDP currently does not intend to work on bundle interfaces.
- LLDP is disabled for the following interfaces on L2BSA L2X:

  Physical interfaces on Q2C platform

  LAG member interfaces on QAX platform

Interactions with other features

RBFS provides various daemons that run as background processes. LLDP interacts with these daemons for accessing necessary configurations.



The table below shows the interactions of LLDP with the other daemons of RBFS.

Daemon Name	Daemon Description	Interaction Details
Confd	Configuration	*Confd* stores all the LLDP configuration information. LLDP subscribes for the following tables of *Confd*: • global.lldp.config • global.lldp.interface.config
IFMD	Interface Monitoring	LLDP is enabled on the physical interface. LLDP subscribes to the IFP table to find the status of the physical link. An LLDP message can be sent only to those interfaces whose link state is UP.
FIBD	Forwarding Information Base	*FIBD* subscribes for the LLDP interface table which has all the information required for enabling LLDP on the interface. This is used to program in VPP. Any neighbor detected, VPP will notify and this detection is stored in the FIBD table for tracking purpose. Each time a message is received from the neighbor, it will be updated in this table. Thus, the history of the messages received from the neighbor is tracked. This information is stored in one of the FIBD tables. LLDP subscribes to this table to find its neighbor.

LLDP Limitations with L2X

LLDP is disabled in the following scenario:

1. The outgoing-interface is configured in the egress L2X.

2. The outgoing-interface is configured in the ingress L2X with the match-type as "any" or "untagged".
3. The outgoing-interface and incoming-interface are configured in the bi-directional L2X.

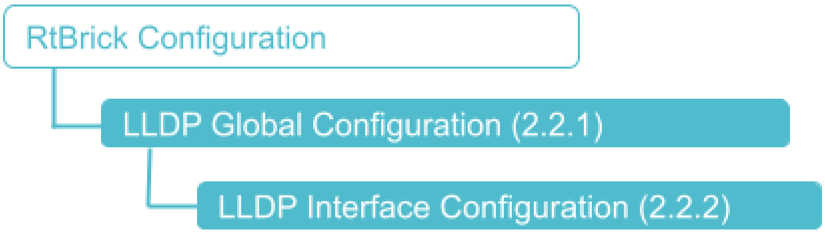
Supported Platforms

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

13.5.2. LLDP Configuration

Configuration Hierarchy

The diagram illustrates the LLDP configuration hierarchy.


RtBrick Configuration
LLDP Global Configuration (2.2.1)
LLDP Interface Configuration (2.2.2)

Configuration Syntax and Commands

The following sections describe the LLDP configuration syntax and commands.

LLDP Global Configuration

Syntax:

set lldp <attribute> <value>

Attribute	Description
admin-status [disable|enable]	Enable or disable LLDP. LLDP is enabled, by default.
system-description <system description>	LLDP global system description to be sent to the neighbor

Attribute	Description
system-name <system name>	LLDP global system name to be sent to the neighbor. If the system name is not configured, it is fetched from BDS.
tx-hold <transmit hold time>	Specifies the amount of time (in seconds) a receiving device maintains the neighbor information before aging your device. If the timer expires and no LLPD packet was received, the neighbor will be marked as DOWN. Default value is 120 seconds. The hold-time range is 1 through 360000.
tx-interval <advertisement interval>	Interval (in seconds) at which LLDP packets are sent to neighbors. Default interval value is 30 seconds. The transmission interval range is 1 through 3600 seconds.

Example 1: Enable LLDP

```
{
    "rtbrick-config:lldp": {
      "admin-status": "enable",
      "system-name": "rtbrick",
      "system-description": "This is rtbrick system",
      "tx-interval": 40,
      "tx-hold": 150
    }
  }
```

LLDP Interface Configuration

Syntax:

set lldp interface <interface-name> <attribute> <value>

Attribute	Description
interface <interface name>	Name of the interface on which you enable or disable LLDP. By default, the interface is enabled. NOTE: Interface level configuration will override the global LLDP enable/disable functionality.

Attribute	Description
admin-status [disable or enable]	LLDP is enabled by default. The command *set lldp admin-status disable* is used to disable it. If you want to re-enable it, run the *set lldp admin-status enable* command.
port-desc <port description>	LLDP port description to be sent to the neighbor. If the port description is not configured, the description configured under interface from IFMD is taken as LLDP port description.

Example 1: LLDP on Interface Configuration

```
{
    "rtbrick-config:lldp": {
      "interface": [
        {
          "interface-name": "ifp-0/0/1",
          "port-description": "this is port ifp-0/0/1",
          "admin-status": "enable"
        }
      ]
    }
  }
```

13.5.3. LLDP Operational Commands

LLDP Show Commands

The LLDP show commands provide detailed information about the LLDP global summary, neighbors, and interfaces.

Syntax:

show lldp <option>

Option	Description
interface	Displays information about interfaces where LLDP is enabled.
neighbor	Displays information about all the neighbors.

Option	Description
summary	Displays global summary details such as system name, frequency of transmissions, the hold-time for the packets sent, TLVs, and the disabled TLVs.

LLDP Interfaces

The command displays the information about interfaces where LLDP is enabled.

Syntax:

show lldp interface <attribute> <value>

Option	Description
interface-name	Displays summary details such as status, MAC address and description.

Example 1: LLDP interface summary

```
root@rtbrick: op> show lldp interface
Interface name       Status  MAC address           Description
ifp-0/0/0                   Down    7a:77:6a:01:00:01   Physical interface #0 from
node 0, chip 0
ifp-0/0/1                   Up         7a:77:6a:01:00:02   Physical interface #1
from node 0, chip 0
ifp-0/0/2                   Up          7a:77:6a:01:00:05   Physical interface #2
from node 0, chip 0
```

Example 2: For the interfaces where L2X is configured, LLDP is suppressed as shown in the following example.

```
root@rtbrick: op> show lldp interface
Interface name Status MAC address Description
ifp-0/0/0 Suppress 7a:3a:ce:60:00:00 Suppressed by L2X : L2X1
ifp-0/0/1 Suppress 7a:3a:ce:60:00:01 Suppressed by L2X : L2X2
```

LLDP Neighbors

This command displays the information of all neighbors.

Syntax:

show lldp neighbor <attribute> <value>

Option	Description
detail	Displays the information in detail about a specific LLDP neighbor or all neighbors.
interface-name	Name of the interface on which neighbor is formed.

Example 1: LLDP neighbor summary

```
root@rtbrick: op> show lldp  neighbor
Neighbor name      Status  Remote port ID     Local port ID      Neighbor MAC
address  Last received     Last sent
fwdd-r2                   Up         ifp-0/0/1                 ifp-0/0/1
7a:1a:c9:00:00:01            0:00:05 ago       0:00:05 ago
fwdd-r2                   Up         ifp-0/0/2                 ifp-0/0/2
7a:1a:c9:00:00:04            0:00:05 ago       0:00:05 ago
```

Example 2: LLDP all neighbor details

```
root@rtbrick: op> show lldp neighbor detail

Neighbor: fwdd-r2
  Neighbor MAC address: 7a:1a:c9:00:00:01
  Neighbor port ID: ifp-0/0/1
  Neighbor port description: Physical interface #1 from node 0, chip 0
  Neighbor TTL: 121
  Neighbor timeout: 121000
  Local interface: ifp-0/0/1
  Local MAC address: 7a:77:6a:01:00:02
  Local port description: Physical interface #1 from node 0, chip 0
  Packets sent: 35
  Packets received: 36
  Neighbor status: Up

Neighbor: fwdd-r2
  Neighbor MAC address: 7a:1a:c9:00:00:04
  Neighbor port ID: ifp-0/0/2
  Neighbor port description: Physical interface #2 from node 0, chip 0
  Neighbor TTL: 121
  Neighbor timeout: 121000
  Local interface: ifp-0/0/2
  Local MAC address: 7a:77:6a:01:00:05
  Local port description: Physical interface #2 from node 0, chip 0
  Packets sent: 35
  Packets received: 36
  Neighbor status: Up
```

Example 3: LLDP specific neighbor details

```
root@rtbrick: op> show lldp neighbor ifp-0/0/1
Neighbor: fwdd-r2
  Neighbor MAC address: 7a:1a:c9:00:00:01
  Neighbor port ID: ifp-0/0/1
```

```
    Neighbor port description: Physical interface #1 from node 0, chip 0
    Neighbor TTL: 121
    Neighbor timeout: 121000
    Local interface: ifp-0/0/1
    Local MAC address: 7a:77:6a:01:00:02
    Local port description: Physical interface #1 from node 0, chip 0
    Packets sent: 148
    Packets received: 149
    Neighbor status: Up
```

LLDP Summary

The command displays the LLDP global summary information.

Syntax:

show lldp summary

Option	Description
-	Without any options, the command displays LLDP global summary details such as status, MAC address and description.

Example 1: LLDP System Summary

```
root@rtbrick: op> show lldp summary
Mode: global
  System hostname: fwdd-r1
  Transmit interval: 30 sec
  Transmit holdtime: 120 sec
```

LLDP Clear Commands

Clear commands allow to reset operational states.

LLDP Neighbor

This commands resets LLDP neighbor.

Syntax:

clear lldp neighbor <option>

Option	Description
all	Clears all the LLDP neighbors.
<ifp-0/0/1>	Clears a specific neighbor on an interface.

Example: The example below shows how to clear all the LLDP neighbors.

```
supervisor@rtbrick: op> clear lldp neighbor all
```

14. Multicast

14.1. IGMP

14.1.1. IGMP Overview

Internet Group Management (IGMP) protocol allows a host to advertise its multicast group membership to neighboring switches and routers. IGMP is a standard protocol used by the TCP/IP protocol suite to achieve dynamic multicasting.

There are two components in the IGMP solution:

- IGMPv2/v3 Client: It sends Join or Leave messages to a multicast group. Typical example of a client is a SET-TOP box. The IGMP client can respond to any IGMP general queries or group-specific queries that are received.
- Multicast Router: The recipient of IGMP Join/Leave message. After receiving the message, it determines whether the corresponding message needs to be processed or not. After processing the IGMP messages, it sends this information to its multicast upstream router. Along with this, it can program certain entries in its routers which results in forwarding specific multicast packets on that interface.

IGMPv3 Lite

IGMP version 3 adds support for "source filtering", that is, the ability for a system to report interest in receiving packets **only** from specific source addresses, or from **all but** specific source addresses, sent to a particular multicast address. That information may be used by multicast routing protocols to avoid delivering multicast packets from specific sources to networks where there are no interested receivers.

The RtBrick IGMP v3lite solution adds support for source filtering. Source filtering enables a multicast receiver host to signal from which groups it wants to receive multicast traffic, and from which sources this traffic is expected. That information may be used by multicast routing protocols to avoid delivering multicast packets from specific sources to networks where there are no interested receivers.

IGMP Version 3 will help conserve bandwidth by allowing a host to select the specific sources from which it wants to receive traffic. Also, multicast routing protocols will be able to make use of this information to conserve bandwidth when constructing the branches of their multicast delivery trees.

Static Joins

After an interface on a multicast device is configured to statically join an IGMP group, the multicast device considers that the interface has static multicast group members and sends multicast packets to this interface, regardless of whether hosts connected to this interface request the multicast packets.

SSM Mapping

SSM mapping takes IGMPv2 reports and converts them to IGMPv3. In case of legacy devices, there could be a possibility that BNG might receive IGMPv2 membership reports. If BNG receives an IGMPv2 membership for a specific group G1, BNG uses the SSM mapping configuration to determine one or more Source (S) addresses for a given group. This SSM mappings are translated to the IGMPv3 joins like IGMPV3 JOIN INCLUDE (G, [S1, G1], [S2, G1] and so on) and BNG continues to process as if it has received from the subscriber.

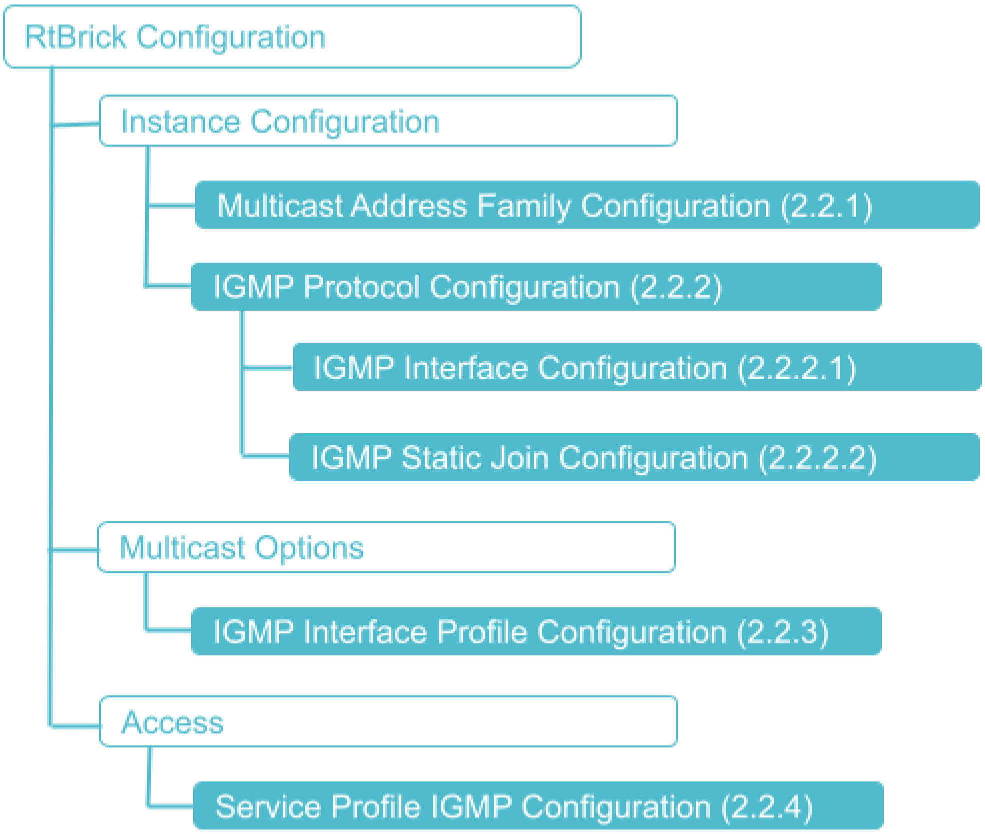
Supported Platforms

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

14.1.2. IGMP Configuration

Configuration Hierarchy

The diagram illustrates the IGMP configuration hierarchy.


RtBrick Configuration
Instance Configuration
Multicast Address Family Configuration (2.2.1)
IGMP Protocol Configuration (2.2.2)
IGMP Interface Configuration (2.2.2.1)
IGMP Static Join Configuration (2.2.2.2)
Multicast Options
IGMP Interface Profile Configuration (2.2.3)
Access
Service Profile IGMP Configuration (2.2.4)

Configuration Syntax and Commands

The following sections describe the interface configuration syntax and commands.

Multicast Address Family Configuration

You can enable the multicast IPv4 address family under the IGMP instance using the following command:

Syntax:

set instance <instance> **address-family** <attribute> <value>

Attribute	Description
<instance>	Specifies the name of the network instance
<afi>	Address family identifier (AFI). Supported value: ipv4
<safi>	Subsequent address family identifier (SAFI), that is, multicast.

Example: Multicast Address Family Configuration

```
{
    "rtbrick-config:address-family": [
      {
        "afi": "ipv4",
        "safi": "multicast"
      }
    ]
  }
```

IGMP Protocol Configuration

To configure an IGMP on an instance, the same instance should be enabled globally with AFI IPv4 and SAFI as both unicast and multicast.

Syntax:

set instance <instance> **protocol igmp** <attribute> <values>

If no instance is specified, IGMP will be enabled on the default instance.

Options	**Description**
<instance>	Name of the IGMP instance.
interfaces <…>	IGMP interface configuration. Refer to section 2.2.1.1 for the IGMP interface configuration.
robustness-variable <variable-value>	The robustness value is used by IGMP to determine the number of times to send messages. Default value: 3. Range: 0-255.
source-address <source-address>	Source address of the IGMP query at the instance-level. NOTE: IF subscriber IFL is configured with the source address, then takes priority; otherwise, the the instance-level source address will be used. If source address is not configured, 0.0.0.0 will be the default address.
static-group <…>	Static multicast route configuration. Refer to section 2.2.2.2 for the IGMP static join configuration.

Example: IGMP Configuration

```
{
  "rtbrick-config:igmp": {
   "robustness-variable": 5,
   "source-address": "198.51.100.177"
  }
 }
```

IGMP Interface Configuration

> When you start IGMP on an interface, it operates with the default settings.

Syntax:

set instance <instance> **protocol igmp interfaces interface** <interface-name> <attribute> <value>

Attribute	Description
<instance>	Name of the instance
<interface-name>	Name of the IP multicast interface
max-groups <count>	Specifies the maximum count of multicast group memberships
version <version>	Specifies the IGMP version, that is, IGMPv2 or IGMPv3
interface-profile <profile>	Name of the interface configuration profile

Example: IGMP Interface Configuration

```
{
  "rtbrick-config:interface": [
   {
    "interface-name": "ifl-0/0/0/1",
    "version": "IGMPv3",
    "max-groups": 30,
    "interface-profile": "profile1"
   }
  ]
 }
```

IGMP Static Join Configuration

Syntax:

set instance <instance> **protocol igmp static-group** <attribute> <value>

Command Parameters

<instance>	Specifies the instance name
<group-address>	Specifies the multicast address
<outgoing-interface>	Name of the outbound interface. The null0 is a discard or sink interface for IGMP static join configuration.

Example: IGMP Static Join Configuration

```
{
    "rtbrick-config:static-group": [
      {
        "group-address": "198.51.100.200",
        "source-address": "198.51.100.1",
        "outgoing-interface": "null0"
      }
    ]
  }
```

IGMP Interface Profile Configuration

Syntax:

set multicast-options igmp interface-profile <attribute> <value>

Attribute	**Description**
filter-policy <filter-policy>	Specifies the filter policy. The policy should be defined under policy statement.
immediate-leave <enable | disable>	Enable or disable the immediate leave option. The immediate-leave attribute removes group membership immediately upon receiving a group leave membership report. If enabled, IGMP perform an immediate leave upon receiving an IGMP group leave message. If the router is IGMP-enabled, it sends an IGMP last member query with a last member query response time. However, the router does not wait for the response time before it prunes off the group querier-timeout-interval IGMP other querier timeout. Default: 425s

Attribute	Description
query-interval <query-interval>	IGMP query interval in seconds. The query interval ranges from 1 to 1024 seconds. The default value is 125 seconds.
query-max-response-time <query-max-response-time>	Maximum query response interval in seconds. The maximum query response interval ranges from 1 to 1024 seconds. The default value is 100 seconds.
ssm-map-policy <ssm-map-policy>	IGMP SSM policy name. The policy for (**,G) mapping to (S,G**)
start-query-count <start-query-count>	Specifies the number of queries sent out on startup, separated by the Start Query Interval. The start query count ranges from 1 to 1024. The default value is 3.
start-query-interval <start-query-interval>	Specifies the start query interval. The start-query-interval ranges from 1 to 1024 seconds. The default value is 31 seconds (query-interval/4).

Example: IGMP Interface Profile Configuration

```
{
    "rtbrick-config:interface-profile": [
      {
        "profile-name": "profile1",
        "immediate-leave": "enable",
        "query-interval": 30,
        "query-max-response-time": 10,
        "start-query-count": 10,
        "start-query-interval": 10,
        "filter-policy": "filter_policy",
        "ssm-map-policy": "ssm_policy"
      }
    ]
  }
```

Service Profile IGMP Configuration

Syntax:

set access service-profile <profile-name> **igmp** <attribute> <value>

Attribute	Description
<profile-name>	Name of the service profile

Attribute	Description
enable <true|false>	Enable IGMP service
max-members <max-members>	Maximum IGMP membership per subscriber
profile <profile>	IGMP profile
version [IGMPv1/IGMPv2/IGMPv3]	IGMP version. The default IGMP version is IGMPv3.

Example: Service Profile IGMP Configuration

```
{
    "rtbrick-config:service-profile": [
      {
        "profile-name": "service-profile1",
        "igmp": {
          "enable": "true",
          "profile": "INTERFACE_PROFILE_1",
          "version": "IGMPv3",
          "max-members": 10
        }
      }
    ]
  }
```

IGMP Configuration Example

```
{
  "ietf-restconf:data": {
    "rtbrick-config:instance": [
      {
        "name": "default",
        "protocol": {
          "igmp": {
            "robustness-variable": 5,
            "source-address": "198.51.100.91"
          }
        }
      }
    ],
    "rtbrick-config:multicast-options": {
      "igmp": {
        "interface-profile": [
          {
            "profile-name": "INTERFACE_PROFILE_1",
            "query-interval": 10,
            "filter-policy": "FILTER_POLICY_1"
          },
          {
            "profile-name": "INTERFACE_PROFILE_2",
            "query-interval": 20,
            "ssm-map-policy": "SSM_POLICY_1"
          }
        ]
      }
    },
    "rtbrick-config:policy": {
      "statement": [
        {
          "name": "FILTER_POLICY_1",
          "ordinal": [
            {
              "ordinal": 1,
              "match": {
                "rule": [
                  {
                    "rule": 1,
                    "type": "ipv4-mcast-group",
                    "value-type": "discrete",
                    "match-type": "or-longer",
                    "value": "198.51.100.20/24"
                  }
                ]
              },
              "action": {
                "rule": [
                  {
                    "rule": 1,
                    "operation": "return-deny"
                  }
                ]
              }
            },
            {
              "ordinal": 2,
              "action": {
                "rule": [
                  {
                    "rule": 1,
                    "operation": "return-permit"
                  }
                ]
              }
            }
          ]
        },
        {
          "name": "SSM_POLICY_1",
          "ordinal": [
            {
              "ordinal": 1,
              "match": {
                "rule": [
                  {
                    "rule": 1,
                    "type": "ipv4-mcast-group",
                    "value-type": "discrete",
                    "match-type": "or-longer",
                    "value": "198.51.100.10/24"
                  }
                ]
              },
              "action": {
                "rule": [
                  {
                    "rule": 1,
                    "type": "ipv4-mcast-source",
                    "operation": "overwrite",
                    "value": "198.51.100.11/24"
                  }
                ]
              }
            }
          ]
        }
      ]
    },
    "rtbrick-config:access": {
      "interface": {
        "double-tagged": [
          {
            "interface-name": "ifp-0/0/1",
            "outer-vlan-min": 1,
            "outer-vlan-max": 4049,
            "inner-vlan-min": 1,
            "inner-vlan-max": 4049,
            "access-type": "PPPoE",
            "access-profile-name": "access-profile1",
            "service-profile-name": "service-profile1",
            "aaa-profile-name": "aaa-profile1"
          }
        ]
      },
      "service-profile": [
        {
          "profile-name": "service-profile1",
          "igmp": {
            "enable": "true",
            "profile": "INTERFACE_PROFILE_1",
            "version": "IGMPv3",
            "max-members": 10
          }
        }
      ]
    }
  }
}
```

14.1.3. IGMP Operational Commands

IGMP Show Commands

Syntax:

show igmp <option>

Option	Description
group	IGMP group summary information
group <group>	IGMP group detailed information
group instance <name>	IGMP group summary information in a specific instance
group outgoing-interface <interface_name>	IGMP group detailed information over a specific interface
interface	IGMP logical-interface summary information
interface instance <name>	IGMP interface summary information on specific instance
interface <interface_name>	IGMP interface detailed information

Example 1: Display IGMP interface details for all instances

```
supervisor@rtbrick>LEAF01: op> show igmp interface
Interface                     Primary Address   State          Querier Address
Instance          Uptime
null0                         n/a               n/a            n/a
vpn1            n/a
ppp-0/0/3/72339069014638597   198.51.100.100     Querier        198.51.100.133
vpn1    03h:37m:31s
```

Example 2: Display the interface summary for a specific instance

```
supervisor@rtbrick>LEAF01: op> show igmp interface instance vpn1
Interface                     Primary Address   State          Querier Address
Instance          Uptime
null0                         n/a               n/a            n/a
vpn1            n/a
ppp-0/0/3/72339069014638597   198.51.100.100     Querier       198.51.100.133
vpn1            03h:37m:39s
```

Example 3: Display IGMP group summary on all instances

```
supervisor@rtbrick>LEAF01: op> show igmp group
Source Address      Group Address       Interface                     Instance
Uptime         Expires     Version
198.51.100.79      198.51.100.233       null0                           vpn1
03h:42m:33s    n/a         IGMP
198.51.100.43   198.51.100.222         null0                           vpn1
03h:42m:33s    n/a         IGMP
198.51.100.51     198.51.100.71         ppp-0/0/3/72339069014638597     vpn1
00h:32m:58s    1m 43s      IGMPv3
```

```
198.51.100.51      198.51.100.72        ppp-0/0/3/72339069014638597     vpn1
00h:33m:03s    1m 48s      IGMPv3
198.51.100.53         198.51.100.73     ppp-0/0/3/72339069014638597     vpn1
00h:35m:26s    3m 36s      IGMPv3
198.51.100.54         198.51.100.74     ppp-0/0/3/72339069014638597     vpn1
00h:35m:26s    3m 35s      IGMPv3
198.51.100.56         198.51.100.115    ppp-0/0/3/72339069014638597     vpn1
03h:38m:16s    3m 33s      IGMPv3
198.51.100.57         198.51.100.117    ppp-0/0/3/72339069014638597     vpn1
03h:38m:16s    3m 29s      IGMPv3
198.51.100.58         198.51.100.18     ppp-0/0/3/72339069014638597     vpn1
03h:38m:16s    3m 42s      IGMPv3
198.51.100.59         198.51.100.19     ppp-0/0/3/72339069014638597     vpn1
03h:38m:16s    3m 35s      IGMPv3
198.51.100.90        198.51.100.64      ppp-0/0/3/72339069014638597     vpn1
03h:38m:16s    3m 40s      IGMPv3
198.51.100.225       198.51.100.68      ppp-0/0/3/72339069014638597     vpn1
00h:35m:26s    3m 40s      IGMPv3
```

Example 4: Display the group summary on specific instance

```
supervisor@rtbrick>LEAF01: op> show igmp group instance vpn1
Source Address      Group Address       Interface                     Instance
Uptime         Expires     Version
198.51.100.79     198.51.100.233        null0                           vpn1
03h:42m:37s    n/a         IGMP
198.51.100.43   198.51.100.233          null0                           vpn1
03h:42m:37s    n/a         IGMP
198.51.100.51       198.51.100.71       ppp-0/0/3/72339069014638597     vpn1
00h:33m:02s    1m 40s      IGMPv3
198.51.100.51       198.51.100.72       ppp-0/0/3/72339069014638597     vpn1
00h:33m:07s    1m 45s      IGMPv3
198.51.100.53       198.51.100.73       ppp-0/0/3/72339069014638597     vpn1
00h:35m:30s    3m 41s      IGMPv3
198.51.100.54       198.51.100.74       ppp-0/0/3/72339069014638597     vpn1
00h:35m:30s    3m 43s      IGMPv3
198.51.100.56       198.51.100.115      ppp-0/0/3/72339069014638597     vpn1
03h:38m:20s    3m 43s      IGMPv3
198.51.100.57       198.51.100.117      ppp-0/0/3/72339069014638597     vpn1
03h:38m:20s    3m 42s      IGMPv3
198.51.100.58        198.51.100.18      ppp-0/0/3/72339069014638597     vpn1
03h:38m:20s    3m 39s      IGMPv3
198.51.100.59         198.51.100.19     ppp-0/0/3/72339069014638597     vpn1
03h:38m:20s    3m 42s      IGMPv3
198.51.100.90        198.51.100.64      ppp-0/0/3/72339069014638597     vpn1
03h:38m:20s    3m 37s      IGMPv3
198.51.100.225       198.51.100.68      ppp-0/0/3/72339069014638597     vpn1
00h:35m:30s    3m 36s      IGMPv3
```

Example 5: Display detailed group information for specific group and source on all instances

```
supervisor@rtbrick>LEAF01: op> show igmp group 198.51.100.233 198.51.100.79
(198.51.100.79, 198.51.100.233)
  Outgoing interface      : null0
  Instance                : vpn1
  Source                   : Static
  State                    : No Members Present
  Version                  : IGMP
  Uptime                   : 03h:42m:54s
  Expires                  : n/a
  Membership interval      : n/a
  Last reporter            : n/a
  Last member query count : n/a
  Last member interval     : n/a
  Retransmit time          : n/a
  Max response time        : n/a
```

Example 6: Display detailed group information for specific group and source on all instances

```
supervisor@rtbrick>LEAF01: op> show igmp group outgoing-interface ppp-
0/0/3/72339069014638597
(198.51.100.51, 198.51.100.71)
  Outgoing interface       : ppp-0/0/3/72339069014638597
  Instance                 : vpn1
  Source                   : Dynamic
  State                    : Members Present
  Version                  : IGMPv3
  Uptime                   : 00h:33m:44s
  Expires                  : 1m 43s
  Membership interval      : 110s
  Last reporter            : 198.51.100.100
  Last member query count : 3
  Last member interval     : 1s
  Retransmit time          : 1s
  Max response time        : 0s
(198.51.100.51, 198.51.100.72)
  Outgoing interface       : ppp-0/0/3/72339069014638597
  Instance                 : vpn1
  Source                   : Dynamic
  State                    : Members Present
  Version                  : IGMPv3
  Uptime                   : 00h:33m:49s
  Expires                  : 1m 48s
  Membership interval      : 110s
  Last reporter            : 198.51.100.100
  Last member query count : 3
  Last member interval     : 1s
  Retransmit time          : 1s
  Max response time        : 0s
(198.51.100.53, 198.51.100.73)
  Outgoing interface       : ppp-0/0/3/72339069014638597
  Instance                 : vpn1
  Source                   : Dynamic
  State                    : Members Present
  Version                  : IGMPv3
  Uptime                   : 00h:36m:12s
  Expires                  : 3m 37s
  Membership interval      : 225s
  Last reporter            : 198.51.100.100
  Last member query count : 3
  Last member interval     : 1s
  Retransmit time          : 1s
  Max response time       : 0s
(198.51.100.54, 198.51.100.74)
  Outgoing interface      : ppp-0/0/3/72339069014638597
  Instance                : vpn1
  Source                  : Dynamic
  State                   : Members Present
  Version                 : IGMPv3
  Uptime                  : 00h:36m:12s
  Expires                 : 3m 41s
  Membership interval     : 225s
  Last reporter           : 198.51.100.100
  Last member query count : 3
  Last member interval    : 1s
  Retransmit time         : 1s
  Max response time       : 0s
(198.51.100.56, 198.51.100.115)
  Outgoing interface      : ppp-0/0/3/72339069014638597
  Instance                : vpn1
  Source                  : Dynamic
  State                   : Members Present
  Version                 : IGMPv3
  Uptime                  : 03h:39m:02s
  Expires                 : 3m 38s
  Membership interval     : 225s
  Last reporter           : 198.51.100.100
  Last member query count : 3
  Last member interval    : 1s
  Retransmit time         : 1s
  Max response time       : 0s
```

Example 7: Display detailed group information for specific group and source on selected instance

```
supervisor@rtbrick>LEAF01: op> show igmp group instance vpn1 198.51.100.117
198.51.100.57
(198.51.100.57, 198.51.100.117)
  Outgoing interface      : ppp-0/0/3/72339069014638597
  Instance                : vpn1
  Source                  : Dynamic
  State                   : Members Present
  Version                 : IGMPv3
  Uptime                  : 03h:39m:31s
  Expires                 : 3m 34s
  Membership interval     : 225s
  Last reporter           : 198.51.100.100
  Last member query count : 3
  Last member interval    : 1s
  Retransmit time         : 1s
  Max response time       : 0s
supervisor@rtbrick>LEAF01: op>
```

IGMP Clear Commands

IGMP Interface

Syntax:

clear igmp interface <attribute> <value>

Option	Description
<interface_name>	Clears the specified IGMP interface
instance <instance> statistics	Clears the IGMP interface statistics for the specified instance.

Example: Clear interface IGMP statistics

```
supervisor@rtbrick>LEAF01: op> clear igmp interface instance default statistics
Interface IGMP statistics were successfully cleared
supervisor@rtbrick>LEAF01: op>
```

IGMP Group

Syntax:

clear igmp group <attribute> <value>

Option	Description
all	Clear all IGMP groups present on all instances
instance <instance>	Clear all IGMP groups present on specific instance
interface <interface_name>	Clear all IGMP groups present on specific interface

Example: Clear all IGMP groups present on all instances

```
supervisor@rtbrick>LEAF01: op> clear igmp group all
IGMP groups were successfully cleared
supervisor@rtbrick>LEAF01: op>
```

14.2. PIM

14.2.1. PIM Overview

c Protocol Independent Multicast (PIM) is a multicast routing protocol that runs over an existing unicast infrastructure. RBFS supports PIM source-specific multicast (SSM). PIM-SSM uses a subset of PIM sparse mode and IGMP version 3 (IGMPv3) to permit a client to receive multicast traffic directly from the source.

PIM SSM builds shortest-path trees (SPTs) rooted at the source immediately because in SSM, the router closest to the interested receiver host is informed of the unicast IP address of the source for the multicast traffic. That is, PIM SSM bypasses the RP connection stage through shared distribution trees, as in PIM sparse mode, and goes directly to the source-based distribution tree.

Internet Protocol Television (IPTV) is a service where digital TV signal data is delivered by using Internet protocol (IP). IPTV service networks typically use PIM-SSM as the multicast routing protocol which has the following characteristics:

- Source specific host membership report for a particular multicast group. IGMPv3 allows a host to describe specific sources from which it would like to receive data.
- PIM shortest path forwarding. Source-specific host report for a particular multicast group and initiating PIM (S,G) joins directly and immediately as result.
- No shared tree forwarding. In order to achieve global effectiveness of SSM, all networks must agree to restrict data forwarding to source trees for some recognized group range. The address range 232.0.0.0/8 has been allocated by IANA for use by source specific multicast (SSM).

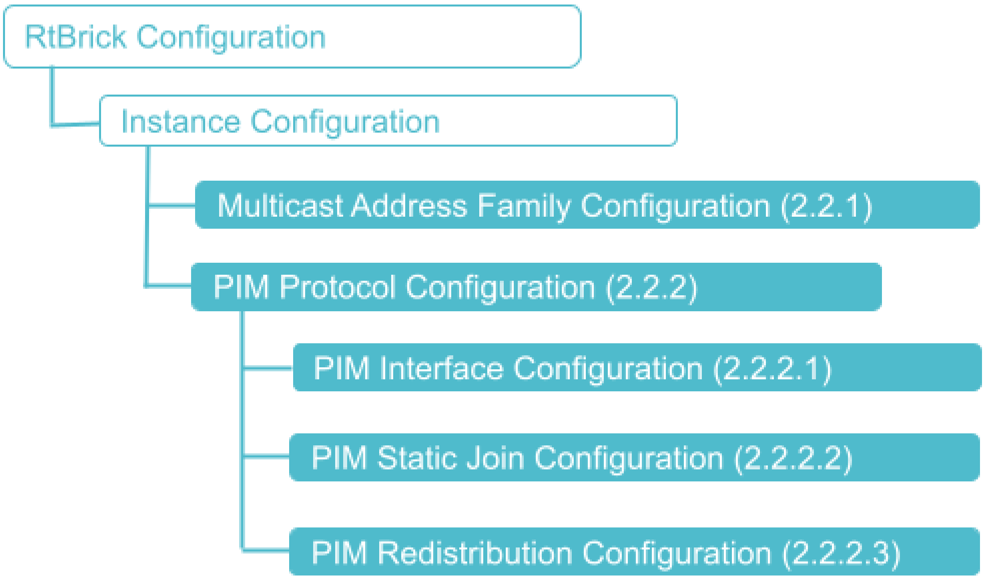
Supported Platforms

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

14.2.2. PIM Configuration

Configuration Hierarchy

The diagram illustrates the PIM configuration hierarchy.


RtBrick Configuration
Instance Configuration
Multicast Address Family Configuration (2.2.1)
PIM Protocol Configuration (2.2.2)
PIM Interface Configuration (2.2.2.1)
PIM Static Join Configuration (2.2.2.2)
PIM Redistribution Configuration (2.2.2.3)

Configuration Syntax and Commands

The following sections describe the interface configuration syntax and commands.

Multicast Address Family Configuration

You can enable the multicast IPv4 address family under the PIM instance using the following command:

Syntax:

set instance <instance> **address-family** <attribute> <value>

Attribute	Description
<instance>	Specifies the name of the network instance
<afi>	Address family identifier (AFI). Supported values: ipv4
<safi>	Subsequent address family identifier (SAFI), that is, multicast.

Example: Multicast Address Family Configuration

```
{
    "rtbrick-config:address-family": [
      {
        "afi": "ipv4",
        "safi": "multicast"
      }
    ]
  }
```

PIM Protocol Configuration

Syntax:

set instance <instance> **protocol pim** <attribute> <value>

If no instance is specified, PIM will be enabled on the default instance. RBFS supports only IPV4 address family.

Attribute	Description
<instance>	Specifies the name of the instance
afi <afi>	Address family identifier (AFI). Supported: ipv4
join-prune-interval <join-prune-interval>	PIM join & prune interval. The interval ranges from 10 to 600 seconds. By default, join & prune interval is 60 seconds and hold-down timer is 210 seconds.
sparse-mode interface <…>	Reference to an entry in the global interface list. Refer to section 2.2.1.1 for the PIM interface configuration.
sparse-mode static-join <…>	A static pim join, (*,G) or (S,G). Refer to section 2.2.1.2 for the PIM static join configuration.
sparse-mode redistribute <…>	Redistribution-related configuration. Refer to section 2.2.1.3 for the PIM redistribution configuration.

PIM Interface Configuration

Syntax:

set instance <instance> **protocol pim sparse-mode interface** <attribute> <value>

Attribute	Description
<instance>	Specifies the name of the instance
<interface-name>	Reference to an entry in the global interface list
afi <afi>	Address family identifier (AFI). Supported: ipv4

Attribute	Description
dr-priority <dr-priority>	Specifies the Specifies the designated router (DR) priority value.
hello-interval <hello-interval>	Specifies the hello timer in seconds. The hello timer ranges from 1 to 3600 seconds. Default: 30 seconds.
override-interval <override-interval>	Specifies the override interval in milliseconds. Default: 2000 milliseconds.
propagation-delay <propagation-delay>	Specifies the propagation delay in milliseconds. Default: 500 milliseconds.

Example: PIM Interface Configuration

```
{
    "rtbrick-config:interface": [
      {
        "interface-name": "ifl-0/0/0/1",
        "hello-interval": 100,
        "dr-priority": 101
      },
      {
        "interface-name": "ifl-0/0/1/2"
      },
      {
        "interface-name": "ifl-0/0/3/3",
        "propagation-delay": 103,
        "override-interval": 1000
      }
    ]
  }
```

PIM Static Join Configuration

Syntax:

set instance <instance> **protocol pim sparse-mode static-join** <attribute> <value>

Attribute	Description
<instance>	Specifies the name of the instance
<interface-name>	Reference to an entry in the global interface list
<outgoing-interface>	Multicast outbound interface name
<group-address>	Multicast group IP address

Attribute	Description
<source-address>	Multicast source IP address

Example: PIM Static Join Configuration

```
{
    "rtbrick-config:static-join": [
      {
        "outgoing-interface": "ifl-0/0/0/1",
        "group-address": "198.51.100.12",
        "source-address": "198.51.100.120"
      }
    ]
  }
```

PIM Redistribution Configuration

Syntax:

set instance <instance> **protocol pim sparse-mode redistribute** <attribute> <value>

Command Parameters

<instance>	Specifies the name of the instance.
<afi>	Specifies the address family identifier. Supported value: ipv4.
<safi>	Specifies the subsequent address family identifier. Supported value: multicast.
<source>	Source protocol from which routes are being redistributed such as BGP or static.

Example: PIM Redistribution Configuration

```
{
    "rtbrick-config:redistribute": [
      {
        "afi": "ipv4",
        "safi": "multicast",
        "source": "bgp"
      }
    ]
  }
```

PIM Configuration Example

```
{
    "rtbrick-config:pim": {
      "sparse-mode": {
        "interface": [
          {
            "interface-name": "ifp-0/0/1"
          }
        ],
        "redistribute": [
          {
            "afi": "ipv4",
            "safi": "multicast",
            "source": "bgp"
          }
        ],
        "static-join": [
          {
            "outgoing-interface": "null0",
            "group-address": "198.51.100.110",
            "source-address": "198.18.73.250"
          },
          {
            "outgoing-interface": "null0",
            "group-address": "198.51.100.111",
            "source-address": "198.18.73.250"
          },
          {
            "outgoing-interface": "null0",
            "group-address": "198.51.100.112",
            "source-address": "198.18.73.250"
          },
          {
            "outgoing-interface": "null0",
            "group-address": "198.51.100.113",
            "source-address": "198.18.73.250"
          }
        ]
      }
    }
  }
```

14.2.3. PIM Operational Commands

PIM Show Commands

PIM Interface

Syntax:

show pim interface <option>

Option	Description
-	Without any option, the commands displays the information for all interfaces.
<interface_name>	Displays the PIM interface detail for the default instance
instance <instance_name>	Displays interface summary on specific instance

Example 1: Summary of PIM logical interface for all instances.

```
supervisor@rtbrick: op> show pim interface
Interface      Instance     IP Address        State      DR                Generator ID
ifl-1/7/1/1    default      198.51.100.247    Non-DR     198.51.100.42     1896236448
null0          default      n/a               n/a        n/a               n/a
ifl-0/0/3/3    vpn1         198.51.100.25     Non-DR     198.51.100.33     2123016228
```

Example 2: Summary of interfaces for the specified instance.

```
supervisor@rtbrick: op> show pim interface instance vpn1
Interface       Instance     IP Address        State      DR                Generator ID
ifl-0/0/3/3     vpn1         198.51.100.25     Non-DR     198.51.100.33     2123016228
```

Example 3: Detailed view of the specified PIM interface.

```
supervisor@rtbrick: op> show pim interface ifl-0/0/3/3
Interface: ifl-0/0/3/3
  Instance                 : vpn1
  State                    : Non-DR
  Primary address          : 198.51.100.25
  Generation ID            : 2123016228
  Timer values
    Hello interval         : 35s
    Join/Prune interval    : 35s
    Hold interval          : 105s
    Override interval      : 2000ms
    Prune delay interval   : 500ms
  DR election
    DR address             : 198.51.100.33
    DR priority            : 1
    DR election count      : 251
  Negotiated
    DR priority used       : True
    Lan delay used         : False
    Lan prune interval     : 0
    Lan override  used     : False
    Lan override interval  : 0
  Statistics
    Hello
      Received             : 17214
      Sent                 : 95
    Membership
      Received             : 0
      Sent                 : 759
    Assert
      Received             : 0
      Sent                 : 0
```

PIM Membership

Syntax:

show pim membership <option>

Option	Description
-	Without any option, the commands displays PIM membership summary on all instances.
instance <instance_name>	Displays the PIM membership summary information on specific instance.
detail	Detailed information on all BGP peers in all instances in a list view.
instance <instance_name> detail	Displays the PIM membership detailed information on specific instance.
<group_address>	Specifies the multicast multicast group address
<source_address>	Specifies the source from which the multicast traffic is received

Example 1: Summary of the PIM membership on all instances.

```
supervisor@rtbrick: op> show pim membership
Source                 Group                 Interface    Instance       Uptime
198.51.100.42          198.51.100.222        null0        default        00h:38m:05s
198.51.100.42          198.51.100.187        null0        default        00h:38m:05s
198.51.100.42          198.51.100.235        null0        default        00h:38m:05s
```

Example 2: Summary of the PIM membership for the specified instance.

```
supervisor@rtbrick: op> show pim membership instance default
Source                 Group                 Interface     Instance       Uptime
198.51.100.42          198.51.100.222        null0         default        00h:38m:18s
198.51.100.42          198.51.100.187        null0         default        00h:38m:18s
198.51.100.42          198.51.100.235        null0         default        00h:38m:18s
```

Example 3: Detailed view of PIM membership on all instances.

```
supervisor@rtbrick: op> show pim membership detail
198.51.100.42, 198.51.100.222
  Instance           : default
  Outgoing interface : null0
  Source             : pim
  Subtype            : Join
  Subsource          : Static
  Uptime             : 00h:39m:29s
198.51.100.42, 198.51.100.187
  Instance           : default
  Outgoing interface : null0
  Source             : pim
  Subtype            : Join
  Subsource          : Static
  Uptime             : 00h:39m:29s
198.51.100.42, 198.51.100.235
  Instance           : default
  Outgoing interface : null0
  Source             : pim
  Subtype            : Join
  Subsource          : Static
  Uptime             : 00h:39m:29s
```

Example 4: Detailed view of PIM membership for the specified instance.

```
supervisor@rtbrick: op> show pim membership instance default detail
198.51.100.42, 198.51.100.222
  Instance           : default
  Outgoing interface : null0
  Source             : pim
  Subtype            : Join
  Subsource          : Static
  Uptime             : 00h:39m:39s
198.51.100.42, 198.51.100.187
  Instance           : default
  Outgoing interface : null0
  Source             : pim
  Subtype            : Join
  Subsource          : Static
  Uptime             : 00h:39m:39s
198.51.100.42, 198.51.100.235
  Instance           : default
  Outgoing interface : null0
  Source             : pim
  Subtype            : Join
  Subsource          : Static
  Uptime             : 00h:39m:39s
```

Example 5: Detailed view of PIM membership for the specified group-address and source-address in selected instance.

```
supervisor@rtbrick: op> show pim membership instance default 198.51.100.222 198.51.100.42
```

```
198.51.100.42, 198.51.100.222
  Instance           : default
  Outgoing interface : null0
  Source             : pim
  Subtype            : Join
  Subsource          : Static
  Uptime             : 00h:39m:50s
supervisor@rtbrick: op>
```

PIM Join and Prune

Syntax:

show pim join-prune <option>

Option	Description
-	Without any option, the commands displays join and prune summary command on all instances.
instance <instance_name>	Displays join and prune summary on a specific instance.
detail	Displays PIM join and prune detailed information on all instances.
instance <instance_name> detail	Displays detailed join and prune information in selected instance.
<group_address>	Displays PIM join and prune detailed information for a specific group on all instance
<source_address>	Specifies the source from which the multicast traffic is received.

Example 1: Summary of the PIM join and prune all instances.

```
supervisor@rtbrick: op> show pim join-prune
Source                  Group              Upstream Interface    Instance
198.51.100.42           198.51.100.222      ifl-1/7/1/1          default
198.51.100.42           198.51.100.187      ifl-1/7/1/1          default
198.51.100.42           198.51.100.235      ifl-1/7/1/1          default
198.51.100.11           198.51.100.22       ifl-0/0/3/3          vpn1
198.51.100.11           198.51.100.217      ifl-0/0/3/3          vpn1
```

Example 2: Summary of the PIM join and prune summary for the specified instance.

```
supervisor@rtbrick: op> show pim join-prune instance vpn1
```

```
Source            Group              Upstream Interface  Instance
198.51.100.11     198.51.100.22      ifl-0/0/3/3         vpn1
198.51.100.11     198.51.100.217     ifl-0/0/3/3         vpn1
198.51.100.33     198.51.100.23      ifl-0/0/3/3         vpn1
198.51.100.40     198.51.100.24      ifl-0/0/3/3         vpn1
```

Example 3: Detailed view of the PIM join and prune for the specified instance.

```
supervisor@rtbrick: op> show pim join-prune instance vpn1 detail
198.51.100.11, 198.51.100.22
  Instance           : vpn1
  Upstream interface : ifl-0/0/3/3
  Upstream neighbor  : 198.51.100.11
  Type               : Join
  Uptime             : 00h:01m:11s
198.51.100.11, 198.51.100.217
  Instance           : vpn1
  Upstream interface : ifl-0/0/3/3
  Upstream neighbor  : 198.51.100.11
  Type               : Join
  Uptime             : 00h:01m:11s
198.51.100.33, 198.51.100.23
  Instance           : vpn1
  Upstream interface : ifl-0/0/3/3
  Upstream neighbor  : 198.51.100.33
  Type               : Join
  Uptime             : 00h:03m:41s
198.51.100.40, 198.51.100.24
  Instance           : vpn1
  Upstream interface : ifl-0/0/3/3
  Upstream neighbor  : 198.51.100.40
  Type               : Join
  Uptime             : 00h:03m:41s
198.51.100.66, 198.51.100.196
  Instance           : vpn1
  Upstream interface : ifl-0/0/3/3
  Upstream neighbor  : 198.51.100.66
  Type               : Join
  Uptime             : 00h:55m:31s
198.51.100.70, 198.51.100.197
  Instance           : vpn1
  Upstream interface : ifl-0/0/3/3
  Upstream neighbor  : 198.51.100.70
  Type               : Join
  Uptime             : 00h:55m:30s
198.51.100.80, 198.51.100.198
  Instance           : vpn1
  Upstream interface : ifl-0/0/3/3
  Upstream neighbor  : 198.51.100.80
  Type               : Join
  Uptime             : 00h:55m:30s
198.51.100.38, 198.51.100.199
  Instance           : vpn1
  Upstream interface : ifl-0/0/3/3
  Upstream neighbor  : 198.51.100.38
  Type               : Join
  Uptime             : 00h:55m:31s
198.51.100.200, 198.51.100.210
  Instance           : vpn1
  Upstream interface : ifl-0/0/3/3
  Upstream neighbor  : 198.51.100.200
  Type               : Join
  Uptime             : 00h:55m:31s
```

Example 4: Detailed view of the PIM join and prune for the specified instance, source, and group.

```
supervisor@rtbrick: op> show pim join-prune instance vpn1 198.51.100.210 198.51.100.200
198.51.100.200, 198.51.100.210
  Instance           : vpn1
  Upstream interface : ifl-0/0/3/3
  Upstream neighbor  : 198.51.100.200
  Type               : Join
  Uptime             : 00h:55m:50s
```

PIM Neighbors

Syntax:

show pim neighbor <option>

Option	Description
-	Without any option, the commands displays the PIM neighbor summary on all instances.
instance <instance_name>	Displays the PIM neighbor summary information on specific instance
instance <instance_name> <neighbor_address>	Displays detailed information for specific PIM neighbor in selected instance.

Example 1: Summary of PIM neighbor on all instances.

```
supervisor@rtbrick: op> show pim neighbor
Neighbor          Interface         Instance      Generation ID  Uptime          Expires
198.51.100.42     ifl-1/7/1/1       default       1413290566     00h:55m:34s     1m 35s
198.51.100.11     ifl-0/0/3/3       vpn1          666648646      00h:55m:59s     21s
198.51.100.37     ifl-0/0/3/3       vpn1          1893441310     00h:55m:59s     28s
198.51.100.33     ifl-0/0/3/3       vpn1          1582670973     00h:55m:56s     24s
198.51.100.40     ifl-0/0/3/3       vpn1          2114142516     00h:55m:59s     21s
198.51.100.50     ifl-0/0/3/3       vpn1          620803409      00h:55m:56s     29s
```

Example 2: Summary of PIM neighbor for the specified instance.

```
supervisor@rtbrick: op> show pim neighbor instance default
Neighbor         Interface           Instance       Generation ID  Uptime         Expires
```

```
198.51.100.42    ifl-1/7/1/1      default        1413290566  00h:55m:41s   1m 28s
```

Example 3: Detailed view of PIM neighbor for the specified instance.

```
supervisor@rtbrick: op> show pim neighbor instance default 198.51.100.42

Neighbor: 198.51.100.42
  Interface            : ifl-1/7/1/1
  Instance             : default
  Hold down interval   : 105s
  Expires              : 105s
  Generation ID        : 1413290566
  DR priority          : 1
  Uptime               : 00h:55m:47s
  Last transition time : Tue Nov 24 06:47:08 GMT +0000 2020
  Holddown received    : 1
```

PIM Reverse Path Forwarding (RPF)

Syntax:

show pim rpf <option>

Option	Description
-	Without any option, the commands displays the PIM rpf summary information on all instance.
instance <instance_name>	Displays the PIM rpf summary information on specific instance.
instance <instance> <source_address>	Displays the PIM rpf detailed information for specific source-address in selected instance.

Example 1: Summary of PIM rpf on all instances.

```
supervisor@rtbrick: op> show pim rpf
Multicast Source  Instance       RPF Interface       Neighbor
198.51.100.42      default        ifl-1/7/1/1      198.51.100.42
198.51.100.11      vpn1           ifl-0/0/3/3      198.51.100.11
198.51.100.33      vpn1           ifl-0/0/3/3      198.51.100.33
198.51.100.40      vpn1           ifl-0/0/3/3      198.51.100.40
198.51.100.66      vpn1           ifl-0/0/3/3      198.51.100.66
198.51.100.107     vpn1           n/a              n/a
```

Example 2: Summary of PIM rpf for the specified instance.

```
supervisor@rtbrick: op> show pim rpf instance vpn1
```

```
Multicast Source  Instance       RPF Interface       Neighbor
198.51.100.11     vpn1         ifl-0/0/3/3      198.51.100.11
198.51.100.33     vpn1         ifl-0/0/3/3      198.51.100.33
198.51.100.40     vpn1         ifl-0/0/3/3      198.51.100.40
198.51.100.66     vpn1         ifl-0/0/3/3      198.51.100.66
198.51.100.70     vpn1         ifl-0/0/3/3      198.51.100.70
198.51.100.107    vpn1         n/a                 n/a
```

Example 3: Detailed view of PIM rpf for the specified source-address in selected instance.

```
supervisor@rtbrick: op> show pim rpf instance vpn1 198.51.100.11
Multicast source : 198.51.100.11
  Instance        : vpn1
  AFI             : ipv4
  SAFI            : unicast
  RPF interface   : ifl-0/0/3/3
  Peer            : 198.51.100.11
  Covering prefix : n/a
  MAC address     : 00:12:01:00:00:01
```

PIM Routes

Syntax:

show pim mroute <option>

Option	Description
-	Without any option, the commands displays the PIM routes summary information on all instance.
detail	Displays the PIM routes detail for all instances.
instance <instance_name>	Displays the PIM routes summary on specific instance
instance <instance_name> detail	Displays the PIM routes detailed information on specific instances.

Example 1: Summary of PIM routes for all instances.

```
supervisor@rtbrick: op> show pim mroute
Instance: default, AFI: ipv4, SAFI: multicast
  Source              Group               Route Source   Preference  Nexthop             OIF
  198.51.100.42      198.51.100.222   pim            240         n/a               null0
  198.51.100.42      198.51.100.187  pim             240         n/a               null0
  198.51.100.42      198.51.100.235  pim             240         n/a               null0
```

Example 2: Detailed view of PIM routes for all instances.

```
supervisor@rtbrick: op> show pim mroute detail
198.51.100.42, 198.51.100.222
  Source       : pim                    Preference       : 240
  Sub source   : Static                 Subtype          : Join
  RPF neighbor : 198.51.100.42          RPF interface    : ifl-1/7/1/1
  Nexthop      : n/a                    Egress interface : null0
  Nexthop type : Multicast Fanout       NextHop action   : None
  Destination  : default-ipv4-multicast
  Resolved in  : default-ipv4-multicast
198.51.100.42, 198.51.100.187
  Source       : pim                    Preference       : 240
  Sub source   : Static                 Subtype          : Join
  RPF neighbor : 198.51.100.42          RPF interface    : ifl-1/7/1/1
  Nexthop      : n/a                    Egress interface : null0
  Nexthop type : Multicast Fanout       NextHop action   : None
  Destination  : default-ipv4-multicast
  Resolved in  : default-ipv4-multicast
198.51.100.42, 198.51.100.235
  Source       : pim                    Preference       : 240
  Sub source   : Static                 Subtype          : Join
  RPF neighbor : 198.51.100.42          RPF interface    : ifl-1/7/1/1
  Nexthop      : n/a                    Egress interface : null0
  Nexthop type : Multicast Fanout       NextHop action   : None
  Destination  : default-ipv4-multicast
  Resolved in  : default-ipv4-multicast
```

Example 3: Summary of PIM routes for the specific instance.

```
supervisor@rtbrick: op> show pim mroute instance default
Instance: default, AFI: ipv4, SAFI: multicast
  Source              Group               Route Source   Preference  Nexthop              OIF
  198.51.100.96    198.51.100.222   pim            240         n/a                 null0
  198.51.100.96    198.51.100.187   pim            240         n/a                 null0
  198.51.100.96    198.51.100.235   pim            240         n/a                 null0
```

Example 4: Detailed view of PIM routes for the specified instances.

```
supervisor@rtbrick: op> show pim mroute instance default detail
198.51.100.96, 198.51.100.222
  Source       : pim                    Preference       : 240
  Sub source   : Static                 Subtype          : Join
  RPF neighbor : 198.51.100.96          RPF interface    : ifl-1/7/1/1
  Nexthop      : n/a                    Egress interface : null0
  Nexthop type : Multicast Fanout       NextHop action   : None
  Destination  : default-ipv4-multicast
  Resolved in  : default-ipv4-multicast
198.51.100.96, 198.51.100.187
  Source       : pim                    Preference       : 240
  Sub source   : Static                 Subtype          : Join
  RPF neighbor : 198.51.100.96          RPF interface    : ifl-1/7/1/1
  Nexthop      : n/a                    Egress interface : null0
  Nexthop type : Multicast Fanout       NextHop action   : None
  Destination  : default-ipv4-multicast
  Resolved in  : default-ipv4-multicast
198.51.100.96, 198.51.100.235
  Source       : pim                    Preference       : 240
```

```
    Sub source   : Static                   Subtype          : Join
    RPF neighbor : 198.51.100.96            RPF interface    : ifl-1/7/1/1
    Nexthop      : n/a                      Egress interface : null0
    Nexthop type : Multicast Fanout         NextHop action   : None
    Destination  : default-ipv4-multicast
    Resolved in  : default-ipv4-multicast
```

PIM Clear Commands

Clear commands allow to reset operational states.

PIM Neighbor

Syntax:

clear pim neighbor

Option	Description
neighbor	Clears all neighbors.

Example: The example below shows how to clear pim neighbor.

```
supervisor@rtbrick: op> clear pim neighbor
All instance clear triggered
```

PIM Neighbor Instance

Syntax:

clear pim neighbor instance <option> ...

Option	Description
neighbor instance <instance_name>	Clears all neighbors on the specified instance.
neighbor instance <instance_name> interface <interface-name>	Clears all neighbors on the specified interface.

Example 1: The example below shows how to clear neighbors of the specified instance

```
supervisor@rtbrick: op> clear pim neighbor instance ip2vrf
Instance clear triggered
```

Example 2: The example below shows how to clear neighbors of the specified interface

```
supervisor@rtbrick: op> clear pim neighbor instance ip2vrf interface ifl-0/0/0
Instance clear triggered
```

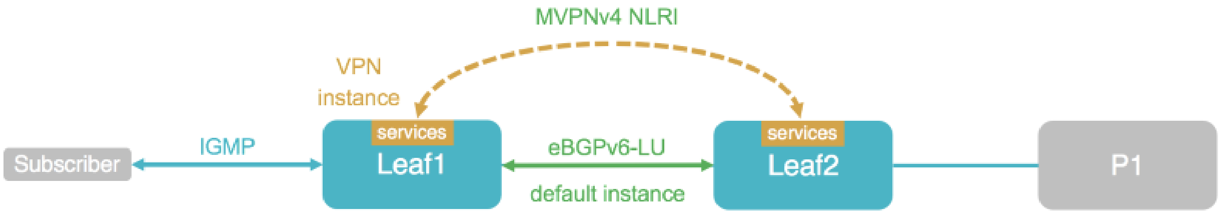
14.3. Multicast VPN

14.3.1. MVPN Overview

The Multicast VPN (MVPN) feature provides the ability to support multicast over a Layer 3 VPN. Multicast allows the efficient distribution of information between a single multicast source and multiple receivers. IP multicast is used to stream video, voice, and data to an MPLS VPN network core. The RBFS MVPN implementation is based on RFC 6513 “Multicast in MPLS/BGP IP VPNs” and RFC 6514 “BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs”.

RFC and draft compliance are partial except as specified.

RBFS can operate in a spine/leaf fabric as shown in the diagram. The leaf device delivers access services to subscribers or assets, and the spine device provides connectivity to the core network. In such a scenario, IPv4 multicast traffic is carried in a multicast VPN instance. This also allows to deliver IPv4 multicast traffic across an IPv6-only fabric. In terms of MVPN, both leaf and spine devices act as Provider Edge (PE) routers.



Multicast subscribers will typically join multicast streams via IGMP. Leaf switches will translate and signal the join messages to the spines using BGP MVPN routes. Spine switches will further forward the join messages towards the source to the upstream core router.

Multicast Address Families

In MVPN, there are two types of instances that address families involved:

- Within the multicast VPN instance, joins are represented as multicast routes. These are AFI 1 (IPv4), SAFI 2 (Multicast). The IPv4 multicast address family needs to be enabled in the VPN service instance as well as for BGP in the VPN instance.
- In the Multicast VPN itself, messages like joins or active sources are advertised using BGP MVPN routes. These are AFI 1 (IPv4), SAFI 5 (MVPN). The BGP peerings that carry the MVPN routes typically run in the default instance. Therefore the MVPN address family needs to be enabled in the default instance, both for the BGP instance itself as well as for the BGP peerings.

In summary, the following address families need to be enabled by configuration:

- The IPv4 multicast address family in the VPN service instance.
- The IPv4 multicast address family for BGP in the VPN service instance.
- The Multicast VPN address family in the BGP default instance.
- The Multicast VPN address family in the BGP peerings in the default instance.

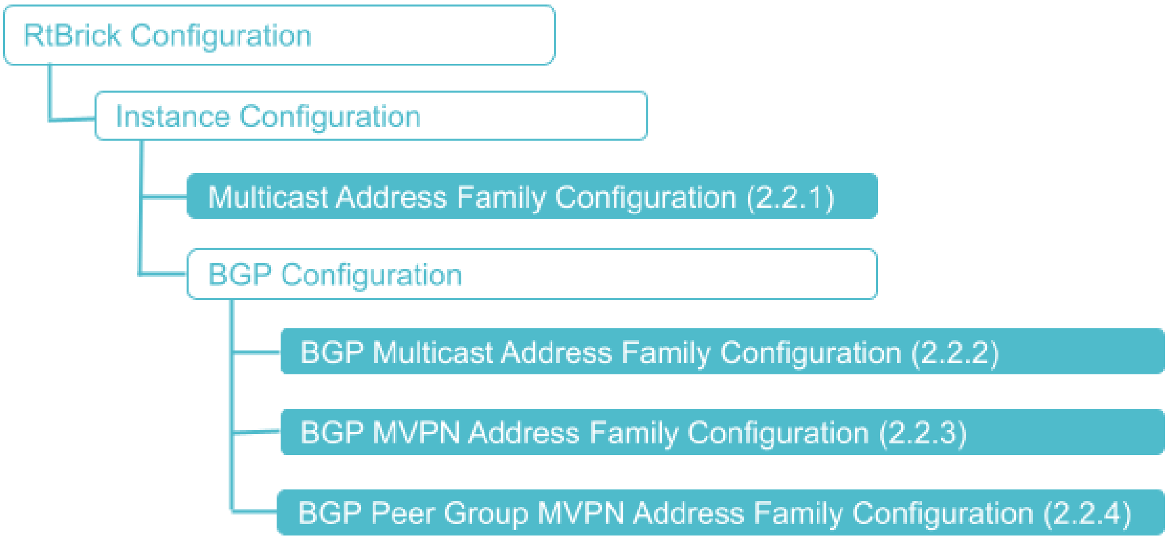
Supported Platforms

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

14.3.2. MVPN Configuration

Configuration Hierarchy

The diagram illustrates the Multicast VPN configuration hierarchy.


RtBrick Configuration
Instance Configuration
Multicast Address Family Configuration (2.2.1)
BGP Configuration
BGP Multicast Address Family Configuration (2.2.2)
BGP MVPN Address Family Configuration (2.2.3)
BGP Peer Group MVPN Address Family Configuration (2.2.4)

Configuration Syntax and Commands

The following sections describe the MVPN configuration syntax and commands.

Multicast Address Family Configuration

Enable the IPv4 multicast address family in the VPN service instance.

Syntax:

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

Attribute	Description
<instance>	The name of the VPN service instance
<afi>	Address family identifier (AFI). For IPv4 multicast, the required value is: ipv4
<safi>	Subsequent address family identifier (SAFI). For IPv4 multicast, the required value is: multicast

Example: Instance Address Family Configuration

```
{
  "ietf-restconf:data": {
    "rtbrick-config:instance": [
      {
        "name": "services",
        "address-family": [
          {
            "afi": "ipv4",
            "safi": "multicast",
            "route-target": {
              "import": "target:198.51.100.10:13",
              "export": "target:198.51.100.10:13"
            }
          }
        ]
      }
    ]
  }
}
```

BGP Multicast Address Family Configuration

Enable the IPv4 multicast address family for BGP in the VPN service instance.

Syntax:

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

Attribute	Description
<instance>	The name of the VPN service instance
<afi>	Address family identifier (AFI). For IPv4 multicast, the required value is: ipv4
<safi>	Subsequent address family identifier (SAFI). For IPv4 multicast, the required value is: multicast
redistribute <source>	Optionally redistribute PIM routes into BGP to translate and advertise subscriber join message as MVPN routes. The required source value is pim as IGMP routes are handled by the PIM process too.

Example: Instance BGP Address Family Configuration

```
{
  "ietf-restconf:data": {
  "rtbrick-config:instance": {
    "name": "services",
    "protocol": {
      "bgp": {
        "address-family": [
          {
            "afi": "ipv4",
            "safi": "multicast",
            "redistribute": [
              {
                "source": "pim"
              }
            ]
          }
        ]
      }
    }
  }
}
```

BGP MVPN Address Family Configuration

Enable the MVPN address family in the global BGP instance.

Syntax:

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

Attribute	Description
<instance>	Name of the global BGP instance. Typically this will be the default instance.
<afi>	Address family identifier (AFI). For MVPN, the required value is: ipv4
<safi>	Supported SAFIs are unicast. For MVPN, the required value is: vpn-multicast

Example: BGP MVPN Address Family Configuration

```
{
  "ietf-restconf:data": {
    "rtbrick-config:instance": [
      {
        "name": "default",
        "protocol": {
          "bgp": {
            "address-family": [
              {
                "afi": "ipv4",
                "safi": "vpn-multicast"
              }
            ]
          }
        }
      }
    ]
  }
}
```

BGP Peer Group MVPN Address Family Configuration

Enable the MVPN address family for the global BGP peerings carrying the MVPN routes.

Syntax:

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

Attribute	Description
<instance>	Name of the global BGP instance. Typically this will be the default instance.
<pg-name>	Peer group name
address-family <afi> <safi>	BGP peer group address family specific. For MVPN, the required AFI/SAFI values are: ipv4/ vpn-multicast.

Example: BGP Peer Group MVPN Address Family Configuration

```
{
  "ietf-restconf:data": {
    "rtbrick-config:peer-group": [
      {
        "pg-name": "spine",
        "address-family": [
          {
            "afi": "ipv4",
            "safi": "vpn-multicast",
            "extended-nexthop": "true",
            "update-nexthop": {
              "ipv6-address": "2001:db8:0:103::"
            }
          }
        ]
      }
    ]
  }
```

Multicast VPN Configuration Example

```
{
  "ietf-restconf:data": {
    "rtbrick-config:instance": [
      {
        "name": "default",
        "protocol": {
          "bgp": {
            "address-family": [
              {
                "afi": "ipv4",
                "safi": "vpn-multicast"
              }
            ],
            "peer-group": [
              {
                "pg-name": "spine",
                "address-family": [
                  {
                    "afi": "ipv4",
                    "safi": "vpn-multicast",
                    "extended-nexthop": "true",
                    "update-nexthop": {
                      "ipv6-address": "2001:db8:0:103::"
                    }
                  }
                ]
              }
            ]
          }
        }
      },
      {
        "name": "services",
        "address-family": [
          {
            "afi": "ipv4",
            "safi": "multicast",
            "route-target": {
              "import": "target:198.51.100.10:13",
              "export": "target:198.51.100.10:13"
            }
          }
        ],
        "protocol": {
          "bgp": {
            "address-family": [
              {
                "afi": "ipv4",
                "safi": "multicast",
                "redistribute": [
                  {
                    "source": "pim"
                  }
                ]
              }
            ]
          }
        }
      }
    ]
  }
}
```

14.3.3. MVPN Operational Commands

Show Commands

Syntax:

show mroute ipv4 instance <vpn-instance-name> <attribute> <value>

Option	Description
-	Without any option, the commands displays summary of all the multicast routes.
<vpn-instance-name>	Name of the VPN instance
detail	Displays detailed view of all the multicast routes.
group <group-address>	Multicast group address
source <source-address>	Multicast source address

Example: Display Multicast Route Summary Information

```
supervisor@rtbrick: op> show mroute ipv4 instance services group 198.51.100.100/24
detail
Instance: services, AFI: ipv4, SAFI: multicast
198.51.100.250/24, 198.51.100.100/24
  Source: pim, Preference: 250
      Next Hop type: Multicast Fanout, Next Hop action: None
      Resolved in: services-ipv4-multicast
      Egress interface: null0
```

MVPN Route Show Commands

Syntax:

show bgp fib ipv4 vpn-multicast instance default <attribute> <value>

Option	Description
-	Without any option, the commands displays summary of all the multicast routes.
detail	Displays detailed view of all the VPN multicast routes.
<prefix>	Destination prefix address

Example: Display MVPN Route Summary Information

```
supervisor@rtbrick: op> show bgp fib ipv4 vpn-multicast instance default
Instance: default, AFI: ipv4, SAFI: vpn-multicast
  Group            Source          Preference      Out Label            Route Type
Next Hop
  -                -                      20       20003,bos:1          Intra-AS_I-
PMSI_AD   -
  -                -                      20       20003,bos:1          Intra-AS_I-
PMSI_AD   -
  198.51.100.111/24    198.51.100.2/24  20       20003,bos:1
Source_Tree_Join     -
```

15. Infrastructure

15.1. NTP

15.1.1. NTP Overview

Network Time Protocol (NTP) provides accurate time across a whole network of devices such as routers or switches and synchronizes time among these devices on a network. NTP uses an NTP server that maintains a highly accurate time. An NTP network is comprised of devices (clients) that are to be synchronized with the NTP server that has UTC time and provides it to the client devices.

Prerequisites

- Install and configure an NTP server on your network and remember the IP address of the NTP server.

Supported Platforms

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

15.1.2. NTP Configuration

Configuration Syntax and Commands

The following sections describe the NTP configuration syntax and commands.

Specifying the NTP Server

To configure the NTP server, you must specify details such as the IPv4 address and the domain name of the NTP server.

Syntax

set system ntp server <server-id> <option>

Attribute	Description
server <server-id>	Specifies the name of the NTP server.
ipv4-address <ipv4-address>	Specifies the IPv4 address of the NTP server.
domain-name <domain-name>	Specifies the domain name of the NTP server.
minpoll	Specifies the minimum interval between requests in seconds. NTP dynamically selects the optimal poll interval between the values specified for minpoll and maxpoll.
maxpoll	Specifies the maximum interval between requests in seconds. NTP dynamically selects the optimal poll interval between the values specified for minpoll and maxpoll.

Example:

```
{
  "ietf-restconf:data": {
    "rtbrick-config:system": {
      "ntp": {
        "server": [
          {
            "server-name": "NtpServerIPv4",
            "ipv4-address": "192.168.16.1",
            "minpoll": 3,
            "maxpoll": 17
          }
        ]
      }
    }
  }
}
```

NTP Service Configuration

You can enable the NTP service under a specific VRF where NTP server is reachable (usually, the inband management VRF).

Syntax:

set inband-management instance <name> **ntp**

Attribute	Description
<name> <instance-name>	Name of the instance
true/false	Enable or disable Network Time Protocol.

Example: NTP Configuration

```
{
    "rtbrick-config:inband-management": {
      "instance": [
        {
          "name": "mgmt-vrf",
          "ntp": "true"
        }
      ]
    }
  }
```

Setting the System Date & Time Using DHCP Option 42 NTP Servers Information

The RtBrick ONL installer image performs a boot time ntpdate operation to set the system date and time.

By default, it uses the pool.ntp.org but DHCP bindings received over the ma1 interface will override it if DHCP option 42 contains a list of one or more NTP servers.

Separately from this boot time setting, the system will continue to use NTPd to maintain accurate system date and time using the list of NTP servers contained in the RBFS configuration.

The /etc/ntp.boot-time-servers.conf file can be used to verify as its contents will be updated with the NTP servers from DHCP. Also, logs are available in /var/log/rtbrick-boot-time-ntpdate.log.

15.1.3. NTP Operational Commands

Verifying NTP Service on Linux

To get more information about NTP’s status, use the ntpq command:

The command displays the NTP servers that the system is synchronized with.

```
supervisor@rtbrick>LEAF01:~ $ sudo ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
198.51.100.5       .XFAC.          16 u    -   64    0    0.000    0.000   0.000
```

15.2. Logging

15.2.1. Logging Overview

RBFS logging is the process of writing log messages during the execution of an event. RBFS logging provides you reports about the events in the entire RBFS ecosystem at different functional areas. Almost every component in an RBFS access network generates a varied form of log files. Therefore, there are many different types of logs. All these logs, which are generated from different components, can be exported to the log management server where you can view and analyze the real-time data. It provides you the ability to trace out the errors of the applications in real-time.

RBFS has been designed based on microservices architecture. An RBFS ecosystem contains multiple microservices and these microservices can be divided as Brick Daemons (BD) and other (non-BD) daemons. CtrlD and ApiGwD daemons are part of ONL, but they do not reside inside of RBFS container. BDS provides in-built infrastructure for logging which can be used by all BDS applications.

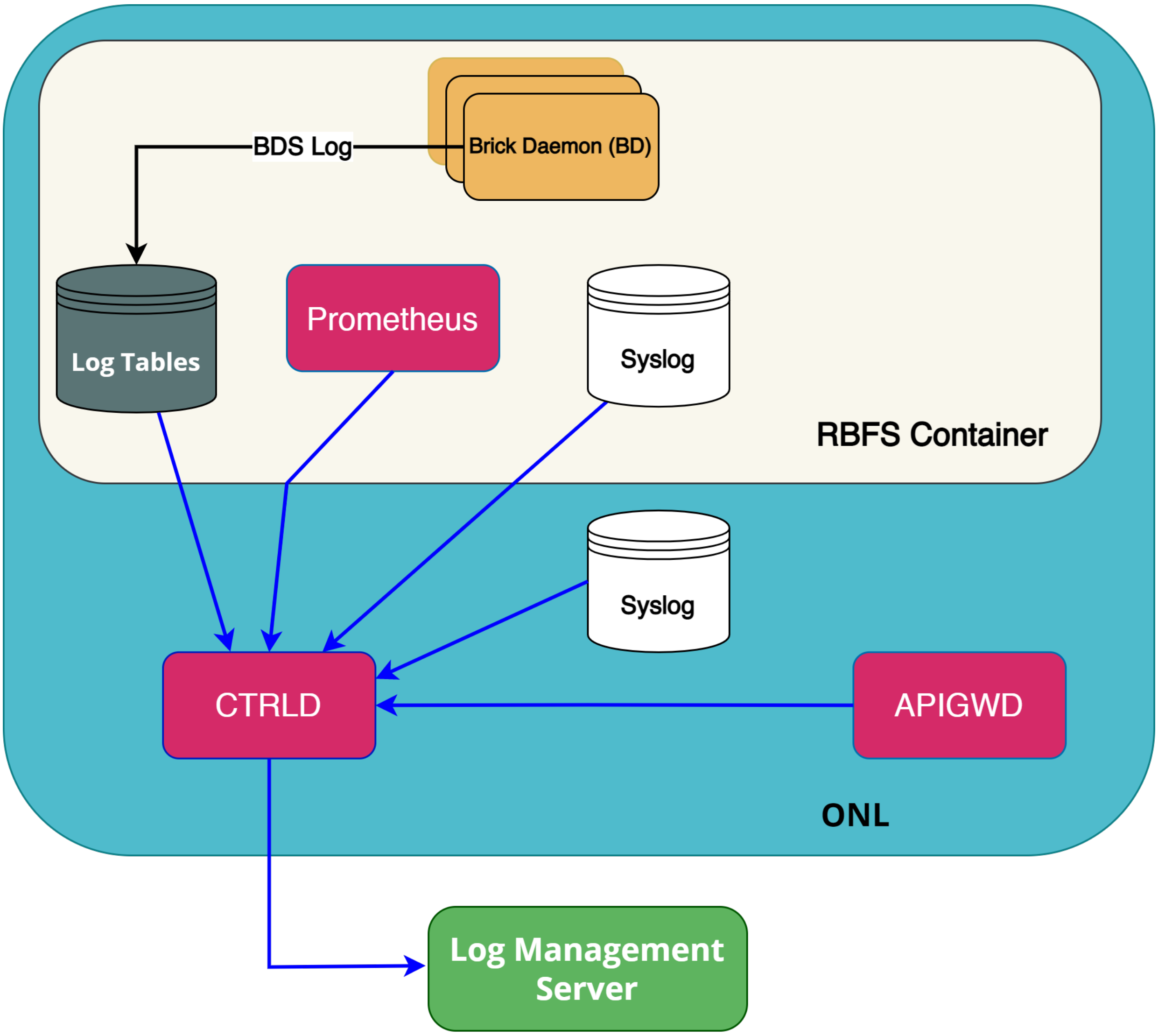
You can configure logging based on different severity levels available.

This document provides you information about logging in the entire RBFS ecosystem. It includes the RBFS container and the host OS which is ONL in case of hardware switches.

Logging in RBFS Container

RBFS provides logging for the entire RBFS ecosystem that includes Brick Daemons (BD) and also for other (non-BD) daemons. Brick Daemons are built on top of BDS and other (non-BD) daemons (such as Prometheus) are the ones which are not dependent on BDS.

The RBFS container logging infrastructure provides in-memory (BDS) and traditional (BD) logging support for RBFS applications. The BDS logging is a low-latency and in-memory logging which can be used in a high scale system without compromising much in performance whereas BD logging is a direct write to a file hence CPU-heavy.

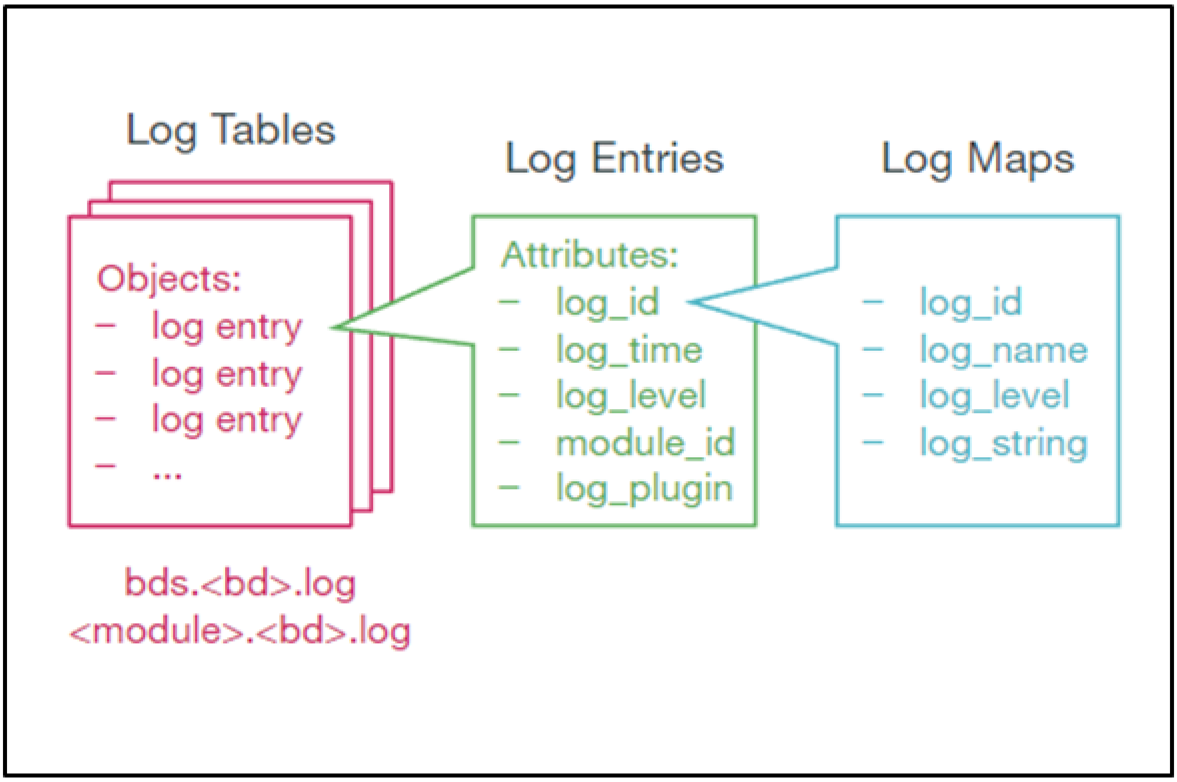


→ The blue arrow indicates logs exported to an external log server

In RBFS, logs are generated from various components or sources such as BDS, Syslog, CTRLD, APIGwD, and Prometheus. All these logs can be finally exported to a log management server.

BDS Logging

BDS logs are stored in a log table. For every unique event, a log ID is created in RBFS. Whenever that particular event is logged, a log entry gets added to the log table. A log table is created for a module only when that module has at least one event logged. Every module in RBFS has at least one log table named in this format: <modulename>.<bd-name>.log.



Log Tables

BDS logs are stored in a BDS table. BDS creates a log table for each module in a BD. One entry is added to this log table for every log. Older entries are removed from the table when the number of entries exceeds more than 10,000.

Log Maps

Every log is mapped to one specific event that is logged by the application. For the optimized usage of memory, RBFS does not store the verbose strings; instead, it stores the log map as an identifier to the actual string message.

Log map and log ID refer to the same entity.

You can access these log maps at the following location:

```
/usr/share/rtbrick/liblog/logs/
```

You can see the log maps, organized based on the modules that they belong to, at:

```
supervisor@rtbrick:/usr/share/rtbrick/liblog/logs$ ls
bds  bgp  fib  fwdinfra  ifm  lldpv2  pd  policy  pubsub  resmon  rib  snapshot
static  time_series
```

In the preceding example, you can see the modules that have registered with the log maps.

If you want to know more about a particular log map, you can perform a *grep* of the log map in this directory.

Log Groups

A log group is a collection of log maps or log IDs. Groups have been introduced to simplify the log configuration tasks. For example, to debug a BGP peer issue, instead of enabling logs for individual log IDs that are related to BGP peer, you can enable log for a log group BGP peer.

Log Modules

Every BDS application consists of multiple modules. Logging can be configured for each BDS modules separately.

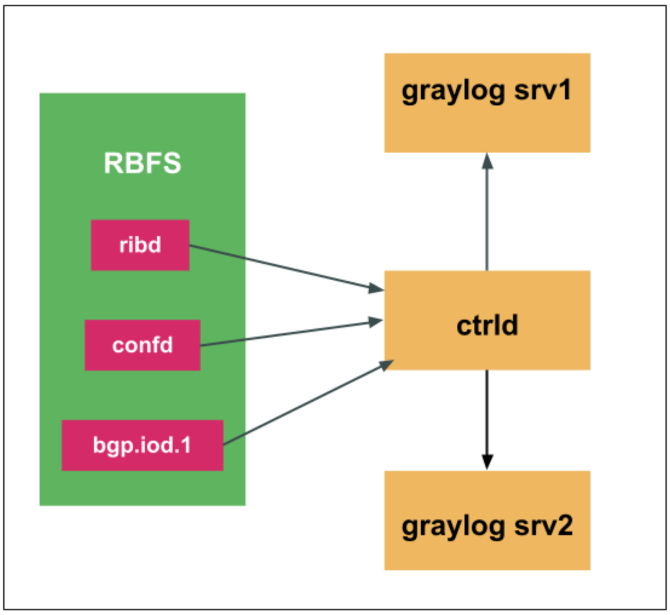
The following are RBFS modules:

```
bcm_q2c
bgp
fib
hostconfd
ifm
igmp
ipoe
isis
l2tp
lag
ldp
license
lldp
mrib
ospf
pim
policy
poold
pppoe
resmon
rib
rtbrick-cli
secure_management
snapshot
subsMgmt
static
time_series
vpp
```

Plugin Alias

Any logs in RBFS can be exported to an external logging destination. Currently, CtrlD supports GELF and syslog as external plugins.

CtrlD is the egress node for all the GELF (Graylog Extended Log Format) messages. The brick daemons which are configured to send GELF messages to CtrlD and CtrlD forwards them to the configured endpoints such as syslog or GELF endpoint. This is because CtrlD enhances the GELF message with switch-global settings (for example, the serial number of the switch).



Guidelines and Limitations for BDS Logging

- By default, BDS logging is enabled and the log level is set to 'Error'.
- By default, logging for BDS and PUBSUB modules have been disabled. As these two modules are infrastructure specific, these logs may not be useful for end-users. However, developers can enable logging for these modules using debug commands.
- You can configure log levels per BD, per module, or per group.
- Do not keep logging enabled for longer duration in a scaled setup.
- The following log levels are present in the system. Any level above the level Warning indicates that you should perform logging with caution as a scaled environment may cause a system instability.

    Emergency

Alert

Critical

Error

Warning

Notice

Info

Debug

None

ℹ If your system becomes unstable, you can remove the logging configuration using the delete log command in configuration mode.

ℹ
- All log levels lower than the log level specified are logged. For example, if the specified log level is "Warning", then all logs that come before "Warning" (Emergency, Alert, Critical, Error, Warning) are logged.
- When you set the log-level to "None", that means log has been disables for the specific module, group, or global.

Syslog

Syslog is generated by an API based logging mechanism provided by Linux. Some of the open source libraries present in RBFS use Syslog as a logging mechanism.

Host operating system and RBFS Linux container use syslog for logging. Syslog can also be exported to Graylog.

Syslog messages can also be exported to CtrlD and these messages are forwarded to the defined log management servers. Currently, RBFS supports exporting syslog from the Linux and ONL system facilities such as auth, authpriv, daemon, and kern to Graylog

For information about Syslog configuration, see section LXC and ONL Logging Configuration.

TSDB (Prometheus) Alert Messages

Prometheus is the systems and service monitoring application, which can be deployed in RBFS, to collect and process metrics. In RBFS, alert messages generated from Prometheus are forwarded to CtrlD and these messages, from CtrlD, can be exported to the configured log management servers.

Logging in ONL

In RBFS, there are daemons, which are not part of RBFS container, but run on the ONL host. RBFS provides logging for these daemons. The following are the daemons which reside on the ONL host:

```
hostconfd
CtrlD
ApiGwD
```

CtrlD and ApiGwD Logs

CtrlD logging provide log messages of events related to business, elements, ZTP, and security. ApiGwD logs contain details about who accessed the API and how they accessed it. hostconfd provides rest APIs to interact with container and ONL linux services.

ApiGwD`and `CtrlD send different log messages about status changes or progress of processes to the configured GELF endpoint.

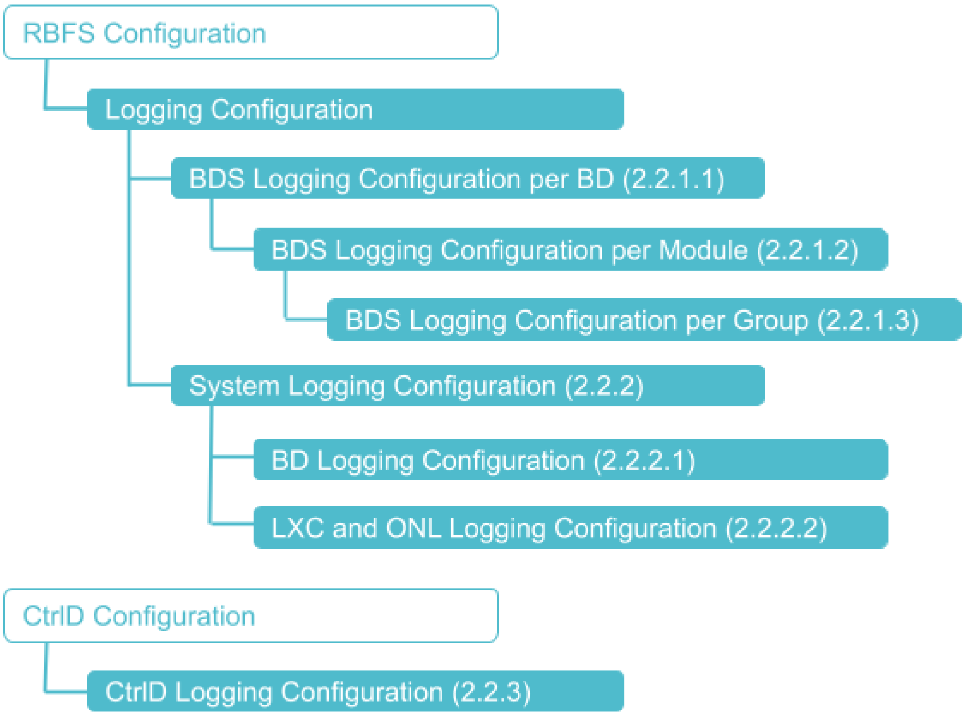
Supported Platforms

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

15.2.2. Logging Configuration

Configuration Hierarchy

The diagram illustrates the logging configuration hierarchy.


RBFS Configuration
Logging Configuration
BDS Logging Configuration per BD (2.2.1.1)
BDS Logging Configuration per Module (2.2.1.2)
BDS Logging Configuration per Group (2.2.1.3)
System Logging Configuration (2.2.2)
BD Logging Configuration (2.2.2.1)
LXC and ONL Logging Configuration (2.2.2.2)
CtrlD Configuration
CtrlD Logging Configuration (2.2.3)

Configuration Syntax and Commands

The following sections describe the logging configuration syntax and commands.

BDS Logging Configuration

You can configure BDS logging for a BD, a module, and for a group.

> A specific configuration takes priority over a generic configuration. For example, if you have configured a global log level of bgp.iod.1 to "warning", and you have configured a log level of bgp module to "notice", then the final log level of bgp will be "notice".

Configuring BDS Logging for a BD

BDS logging can be configured for a BD.

Syntax:

set log bd <bd-name> <option> …

Attribute	Description
<bd-name>	Configure for the specified BD name.

Attribute	Description
all	Configure for all BDs.
module <module-name>	Module name. For more information, see the section "Configuring BDS Logging for a Module".
plugin-alias alias-name <alias-name>	Plugin alias name
plugin-alias level <level>	Log severity level. You can filter logs based on the log severity levels for sending to the external log management server. This will help you to send only the required log messages to the log management server instead of sending a whole lot of data. For example, if you configure Warning as the severity level, logs with the severity level 'Warning' and all the log levels above Warning such as Error, Critical, Alert, and Emergency will be sent to the log management server.

Example 1: BDS logging for a BD Configuration

```
{
  "ietf-restconf:data": {
    "rtbrick-config:system": {},
    "rtbrick-config:log": {
      "bd": [
        {
          "bd-name": "bgp.iod.1",
          "level": "info",
          "plugin-alias": {
            "alias-name": "ztp",
            "level": "error"
          }
        }
      ]
    }
```

Configuring BDS Logging for a Module

Logging can be configured for a module such as BGP, IS-IS, and so on.

Syntax:

set log module <module-name> <option> ...

Attribute	Description
<module-name>	Module name
group <group-name>	BDS log module log-group configuration. For more information, see the section "Configuring BDS Logging for a Group".
level <level>	Log severity level.
plugin-alias alias-name <alias-name>	Plugin alias name
plugin-alias level <level>	Log level. You can filter logs based on the log severity levels for sending to the log management server. This will help you to send only the required log messages to the log management server instead of sending a whole lot of data. For example, if you configure Warning as the severity level, logs with the severity level 'Warning' and all the log levels above Warning such as Error, Critical, Alert, and Emergency will be sent to the log management server.

Example 1: Logging Module Configuration

```
{
  "ietf-restconf:data": {
    "rtbrick-config:system": {},
    "rtbrick-config:log": {
      "module": [
        {
          "module-name": "igmp",
          "level": "info",
          "plugin-alias": {
            "alias-name": "graylog-srv1",
            "level": "warning"
          }
        }
      ]
    }
  }
}
```

Configuring BDS Logging for a Group

Logging can be configured at the group hierarchy level.

Syntax:

set log module <module-name> **group** <group-name> <option> ...

Attribute	Description
<group-name>	Group name
level <level>	Specifies the level of the plug-in alias.
rate-limit <rate-limit>	Rate-limiting is only supported for log groups. Configuring a higher rate-limit for a whole module may cause system instability due to generation of high volume of logs. The default value is 10.
plugin-alias alias-name <alias-name>	Plugin alias name
plugin-alias level <level>	Specifies the level of the plug-in alias. You can filter logs based on the log severity levels for sending to the log management server. This will help you to send only the required log messages to the log management server instead of sending a whole lot of data. For example, if you configure Warning as the severity level, logs with the severity level 'Warning' and all the log levels above Warning such as Error, Critical, Alert, and Emergency will be sent to the log management server.

Example 1: Logging Group Configuration

```
{
  "ietf-restconf:data": {
    "rtbrick-config:log": {
      "module": [
        {
          "module-name": "bgp",
          "group": [
            {
              "group-name": "interface",
              "level": "warning"
            }
          ]
        }
      ]
    }
  }
}
```

System Logging Configuration

LXC and ONL Logging Configuration

You can configure an external log management server to transport logs for real-time analysis. Currently, Syslog and Graylog are the log management servers supported by RBFS.

You can send logs from Linux and ONL system facilities such as auth, authpriv, daemon, and kern to the log management server.

Syntax:

set log system facility <facility-name> **plugin-alias** <attribute> <value>

Attribute	Description
<facility-name>	Supported facilities include all, auth, authpriv, daemon, and kern.
plugin-alias alias-name <alias-name>	Plugin alias name
plugin-alias level <level>	Specify the level of the plug-in alias. You can filter logs based on the log severity levels for sending to the log management server. This will help you to send only the required log messages to the log management server instead of sending a whole lot of data. For example, if you configure Warning as the severity level, logs with the severity level 'Warning' and all the log levels above Warning such as Error, Critical, Alert, and Emergency will be sent to the log management server.

Example 1: System Log Configuration

```
{
  "ietf-restconf:data": {
    "rtbrick-config:log": {
      "system": {
        "facility": [
          {
            "facility-name": "kern",
            "plugin-alias": {
              "alias-name": "graylog",
              "level": "notice"
            }
          }
        ]
      }
    }
  }
}
```

External Log Server (Plugin Alias) Configuration

You can configure an external log management server as a destination to transport logs.

Syntax:

set system host log-alias <attribute> <value>

Attribute	Description
<name>	Log alias name
<address>	Plugin address. For example, Gelf: http://11.1.1.1:1102/gelf Syslog: 10.1.1.1:8008
level	Severity level for the log.
network	Log alias endpoint port.
type	Server type of log alias.

Example 1: Log alias Configuration

```
set system host log-alias bng endpoint http://10.1.1.1:1102/gelf
set system host log-alias bng endpoint http://10.1.1.1:1102/gelf type gelf
set system host log-alias bng endpoint http://10.1.1.1:1102/gelf network http
set system host log-alias bng endpoint http://10.1.1.1:1102/gelf level info
{
  "ietf-restconf:data": {
    "rtbrick-config:system": {
      "host": {
        "log-alias": [
          {
            "name": "bng",
            "endpoint": [
              {
                "address": "http://10.1.1.1:1102/gelf",
                "type": "gelf",
                "network": "http",
                "level": "info"
              }
            ]
          }
        ]
      }
    }
  }
}
```

CtrlD Logging Configuration

When you configure a plugin alias in RBFS, the log message is forwarded to CtrlD. CtrlD forwards it to the corresponding the log management server endpoint alias that you configured.

You must add the log management server endpoint in the CtrlD start-up configuration before configuring a plugin-alias in RBFS.

> If the configured plugin-alias name does not match any of the log management server endpoint name configured in CtrlD, the log is sent to the default the log management server endpoint ("graylog_url").

For Graylog support, you need to configure logging in the CtrlD.

RBFS logs can be sent to the log management server servers. This can be achieved by configuring a plugin alias in CtrlD.

The following section describe the tasks to be performed to configure the plugin alias in CtrlD:

config.json

This section describes the main configuration file of CtrlD. This file can be changed via API. If it is changed on the file system, CtrlD has to be restarted.

/etc/rtbrick/ctrld/config.json example

```
{
  "graylog_enable": true,
  "graylog_url": "http://198.51.100.149:12201/gelf",
  "graylog_endpoints": [
    {
      "name": "ztp",
      "url": "http://198.51.100.146:12201/gelf"
      }
    ]
  }
```

/etc/rtbrick/ctrld/config.json format

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><b>graylog_enable</b></td><td>bool</td><td>To Enable all Graylog outgoing messages</td></tr>
<tr><td><b>graylog_url</b></td><td>string</td><td>Graylog url e.g. http://198.51.100.11:12201/gelf</td></tr>
<tr><td><b>graylog_heart_beat_interval</b></td><td>string</td><td>Graylog heartbeat Interval in seconds (0 means deactivated)</td></tr>
<tr><td><b>graylog_endpoints</b></td><td colspan="2">GraylogEndpoints allows to specify multiple graylog endpoints by name. If a log to a specific endpoint is requested and the endpoint is not available, the log is send to the default Graylog endpoint.
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><b>name</b></td><td>string</td><td>Logical name of the entpoint e.g.: ztp for ztp messages.</td></tr>
<tr><td><b>url</b></td><td>string</td><td>Graylog url e.g. http://198.51.100.11:12201/gelf</td></tr>
<tr><td><b>disable</b></td><td>string</td><td>Disables this endpoint.</td></tr>
</table>
If the default endpoint is disabled, but the specific one is enabled than the message to the specific endpoint will be sent.
If the default endpoint is enabled, but the specific one is disabled than the message to the specific endpoint will not be sent.</td></tr>
</table>

15.2.3. Logging Operational Commands

The logging operational commands provide information about the logging operations. They are used to show logs in the system, log configuration status and so on.

BDS Logging

The BDS logging show commands provide information about the BDS logging operations. With the BDS logging operational commands, you can verify BDS logging configuration status and view BDS logs.

Verifying BDS Logging Configuration Status

Show Log Status

This command shows log configuration status for all modules except infrastructure modules in all BDs. The default show log status displays the summary of log status for the whole system and there are options available to show specific module or BD log status.

Syntax:

show log status <attribute> <value>

Option	Description
-	Without any option, the command displays the log configuration status for all modules except infrastructure modules in all BDs.
bd <bd-name>	Displays log status of the all modules including infrastructure in the specified BD.
detail	Displays the log configuration status in detail all the way till log IDs.
module <module-name>	Displays the log status for a given module in all BDs where this module is running.

Example 1: Summary of log status

```
supervisor@rtbrick: cfg> show log status
Module log status:
  bds_mock:
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level   Rate limit
        generic                  error          None                     none           10
  bgp:
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level   Rate limit
        config                   error          None                     none           10
        general                  error          None                     none           10
        generic                  error          None                     none           10
        instance                 error          None                     none           10
        interface                warning        None                     none           10
        message                  error          None                     none           10
        peer                     error          None                     none           10
    bgp.appd.1:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level   Rate limit
        config                   error          None                     none           10
        general                  error          None                     none           10
        generic                  error          None                     none           10
        instance                 error          None                     none           10
        interface                warning        None                     none           10
        message                  error          None                     none           10
        peer                     error          None                     none           10
    bgp.iod.1:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level   Rate limit
        config                   error          None                     none           10
        general                  error          None                     none           10
        generic                  error          None                     none           10
        instance                 error          None                     none           10
        interface                warning        None                     none           10
        message                  error          None                     none           10
        peer                     error          None                     none           10
  fib:
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level   Rate limit
        adjacency                error          None                     none           10
        bds                      error          None                     none           10
        config                   error          None                     none           10
        general                  error          None                     none           10
        generic                  error          None                     none           10
        instance-afi-safi        error          None                     none           10
        interface-events         error          None                     none           10
        route                    error          None                     none           10
  fwdinfra:
    ribd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level   Rate limit
        generic                  error          None                     none           10
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level   Rate limit
        generic                  error          None                     none           10
    staticd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level   Rate limit
        generic                  error          None                     none           10
    ifmd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level   Rate limit
        generic                  error          None                     none           10
    mribd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level   Rate limit
        generic                  error          None                     none           10
  hostconfd:
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level   Rate limit
        bds                      error          None                     none           10
        config                   error          None                     none           10
        functional               error          None                     none           10
        generic                  error          None                     none           10
  ifm:
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level   Rate limit
        bds                      error          None                     none           10
        config                   error          None                     none           10
        general                  error          None                     none           10
        generic                  error          None                     none           10
        interface-events         error          None                     none           10
    ifmd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level   Rate limit
        bds                      error          None                     none           10
        config                   error          None                     none           10
        general                  error          None                     none           10
        generic                  error          None                     none           10
        interface-events         error          None                     none           10
  igmp:
    pim.appd.1:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level   Rate limit
        config                   error          None                     none           10
        generic                  error          None                     none           10
        igmp-interface-events    error          None                     none           10
        igmp-membership-events   error          None                     none           10
        igmp-packet-events       error          None                     none           10
        igmp-route-events        error          None                     none           10
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level   Rate limit
        config                   error          None                     none           10
        generic                  error          None                     none           10
        igmp-interface-events    error          None                     none           10
        igmp-membership-events   error          None                     none           10
        igmp-packet-events       error          None                     none           10
        igmp-route-events        error          None                     none           10
    igmp.appd.1:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level   Rate limit
        config                   error          None                     none           10
        generic                  error          None                     none           10
        igmp-interface-events    error          None                     none           10
        igmp-membership-events   error          None                     none           10
        igmp-packet-events       error          None                     none           10
        igmp-route-events        error          None                     none           10
    pim.iod.1:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level   Rate limit
        config                   error          None                     none           10
        generic                  error          None                     none           10
        igmp-interface-events    error          None                     none           10
```

Example 2: View of module log status

```
supervisor@rtbrick: cfg>  show log status module bgp
Module log status:
  bgp:
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level   Rate limit
        config                   error          None                     none           10
        general                  error          None                     none           10
        generic                  error          None                     none           10
        instance                 error          None                     none           10
        interface                warning        None                     none           10
        message                  error          None                     none           10
        peer                     error          None                     none           10
    bgp.appd.1:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level   Rate limit
        config                   error          None                     none           10
        general                  error          None                     none           10
        generic                  error          None                     none           10
        instance                 error          None                     none           10
        interface                warning        None                     none           10
        message                  error          None                     none           10
        peer                     error          None                     none           10
    bgp.iod.1:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin                   Plugin Level   Rate limit
        config                   error          None                     none           10
        general                  error          None                     none           10
        generic                  error          None                     none           10
        instance                 error          None                     none           10
        interface                warning        None                     none           10
        message                  error          None                     none           10
        peer                     error          None                     none           10
```

Example 3: Log status of the all modules including infrastructure in the specified BD.

```
supervisor@rtbrick: cfg> show log status bd bgp.appd.1
System/File log status:
  Level: error

Module log status:
  bd:
    Level: none, Plugin: None, Plugin Level: none
    Log group status:
      Group                     Level          Plugin                   Level          Rate limit
      generic                   none           None                     none           10
      http                      none           None                     none           10
  bds:
    Level: none, Plugin: None, Plugin Level: none
    Log group status:
      Group                     Level          Plugin                   Level          Rate limit
      generic                   none           None                     none           10
      object                    none           None                     none           10
      table                     none           None                     none           10
      trim                      none           None                     none           10
  bgp:
    Level: error, Plugin: None, Plugin Level: none
    Log group status:
      Group                     Level          Plugin                   Level          Rate limit
      config                    error          None                     none           10
      general                   error          None                     none           10
      generic                   error           None                      none           10
      instance                  error           None                      none           10
      interface                 warning         None                      none           10
      message                   error           None                      none           10
      peer                      error           None                      none           10
  license:
    Level: error, Plugin: None, Plugin Level: none
    Log group status:
      Group                     Level           Plugin                    Level          Rate limit
      generic                   error           None                      none           10
      internal                  error           None                      none           10
      operational               error           None                      none           10
  policy:
    Level: error, Plugin: None, Plugin Level: none
    Log group status:
      Group                     Level           Plugin                    Level          Rate limit
      BDS                       error           None                      none           10
      Configuration             error           None                      none           10
      Enforcement               error           None                      none           10
      Generation                error           None                      none           10
      generic                   error           None                      none           10
  poold:
    Level: error, Plugin: None, Plugin Level: none
    Log group status:
      Group                     Level           Plugin                    Level          Rate limit
      generic                   error           None                      none           10
  pubsub:
    Level: none, Plugin: None, Plugin Level: none
    Log group status:
      Group                     Level           Plugin                    Level          Rate limit
      generic                   none            None                      none           10
  secure_management:
    Level: error, Plugin: None, Plugin Level: none
    Log group status:
      Group                     Level           Plugin                    Level          Rate limit
      generic                   error           None                      none           10
  snapshot:
    Level: error, Plugin: None, Plugin Level: none
    Log group status:
      Group                     Level           Plugin                    Level          Rate limit
      generic                   error           None                      none           10
  time_series:
    Level: error, Plugin: None, Plugin Level: none
    Log group status:
      Group                     Level           Plugin                    Level          Rate limit
      generic                   error           None                      none           10
```

Example 4: Log status for given module in the given BD

```
supervisor@rtbrick: cfg>  show log status module bgp bd bgp.appd.1
Module log status:
  bgp:
    bgp.appd.1:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                     Level          Plugin                   Plugin Level   Rate limit
        config                    error          None                     none           10
        general                   error          None                     none           10
        generic                   error          None                     none           10
        instance                  error          None                     none           10
        interface                 warning        None                     none           10
        message                   error          None                     none           10
        peer                      error          None                     none           10
```

Example 5: Log status for active logs per log ID

```
supervisor@rtbrick: cfg> show log status bd bgp.appd.1 detail
System/File log status:
  Level: error

Module log status:
  bd:
    Level: none, Plugin: None, Plugin Level: none
    Log group status:
      generic, Level: none, Plugin: None, Plugin Level: none, Rate limit: 10
      http, Level: none, Plugin: None, Plugin Level: none, Rate limit: 10
     Log ID status:
        LOG ID                                                                Level          Plugin
Level          Rate limit
        HTTP_JWK_FILE_JSON_PARSE_FAILED                                       none           None
none           10
        HTTP_JWK_FILE_MEM_ALLOC_FAILED                                        none           None
none           10
        HTTP_JWK_FILE_MISSING                                                 none           None
none           10
        HTTP_JWK_FILE_OPEN_FAILED                                             none           None
none           10
        HTTP_JWK_FILE_READ_FAILED                                             none           None
none           10
        HTTP_JWK_MISSING_KEY                                                  none           None
none           10
        HTTP_JWK_MULTIPLE_KEYS                                                none           None
none           10
        HTTP_SEND                                                             none           None
none           10
        HTTP_WRITE_BUFFER_MEM_ALLOC_FAILED                                    none           None
none           10
        HTTP_WRITE_PRINTF_FAILED                                              none           None
none           10
  bds:
    Level: none, Plugin: None, Plugin Level: none
    Log group status:
      generic, Level: none, Plugin: None, Plugin Level: none, Rate limit: 10
     Log ID status:
        LOG ID                                                                Level          Plugin
Level          Rate limit
        BDS_ATTRIBUTE_TEMPLATE_EVENT                                          none           None
none           10
        BDS_INVALID_PARAMS                                                    none           None
none           10
        BDS_PUBSUB_ERROR_STATUS                                               none           None
none           10
        BDS_QUEUE_TABLE                                                       none           None
none           10
        BDS_ROOT_EVENT                                                        none           None
none           10
        BDS_TEST_LOG                                                          none           None
none           10
      object, Level: none, Plugin: None, Plugin Level: none, Rate limit: 10
     Log ID status:
        LOG ID                                                                Level          Plugin
Level          Rate limit
```

Viewing BDS Logs

Show Log

This command shows BDS logs in the log tables. By default, the command show log shows all logs present in the log tables. Various command options are available to filter and display logs. Also, a command option is available to send logs into a file.

Syntax:

show log <option>

Option	Description
-	Without any option, the command displays all logs present in the log tables.
filter <level> <module>	Render output of the log can be filtered for the specified module or the log level. You can specify filter level or module with any of the view log command, so that the logs are filtered based on the specified level or module.
format <abstract |summary |detailed>	Choose any of the three formats: abstract, summary, or detailed. This command provides output in the specified output format. You can specify any of the options at end of any of the view log commands to show logs in a specific output format. By default, summary is the log format. Abstract: Shows logs without metadata Detailed: Shows logs in detail with metadata Summary: Shows logs in summary view with metadata
table <name>	Displays logs from a specified log table. Every BD includes multiple log tables. By default, log will be rendered from every log table, if not specified.
to file <filename>	Name of the file in which logs are transported. You can specify to file and the file name at end of any view log command to transport the log to the file.

Example 1: View of logs

```
supervisor@rtbrick: op> show log
[  Error   ] <2021-07-09T04:35:53.184694+0000> Table [global.hostconfd.table.config] - event Failed to open
file
[  Error   ] <2021-07-09T04:35:53.184771+0000> Table [global.hostconfd.table.config] - event Could not create
snapshot block
[  Error   ] <2021-07-09T04:35:53.184849+0000> Table [global.hostconfd.table.config] - event Failed to open
file
[  Error   ] <2021-07-09T04:35:53.184866+0000> Table [global.hostconfd.table.config] - event Could not create
snapshot block
[  Error   ] <2021-07-09T04:35:53.201029+0000> Table [global.time-series.config] - event Failed to open file
[  Error   ] <2021-07-09T04:35:53.201052+0000> Table [global.time-series.config] - event Could not create
snapshot block
```

```
[  Error   ] <2021-07-09T04:35:53.201106+0000> Table [global.time-series.config] - event Failed to open file
[  Error   ] <2021-07-09T04:35:53.201125+0000> Table [global.time-series.config] - event Could not create
snapshot block
[  Error   ] <2021-07-09T04:35:53.222660+0000> Table [secure.global.system.table.config] - event Failed to
open file
[  Error   ] <2021-07-09T04:35:53.222679+0000> Table [secure.global.system.table.config] - event Could not
create snapshot block
[  Error   ] <2021-07-09T04:36:00.720574+0000> Table [global.tacacs.config] Object [name -
tacacs_config_object] attribute - tacacs_server_ip not found event TACACS Server Hostconfd Config
```

Example 2: Summary view for the show log table

```
supervisor@rtbrick: op> show log table secure_management.confd.log
[  Error   ] <Tue Nov 10 19:44:48 GMT +0000 2020> Table [global.tacacs.config] Object [name -
tacacs_config_object] attribute - tacacs_server_ip not found event TACACS Server Hostconfd Config
```

Example 3: View of applied filters on all logs from a single table

```
supervisor@: op> show log table rtbrick-cli.confd.log filter level Info
[   Info   ] <Thu Nov 12 11:20:29 GMT +0000 2020> Commit Success
[   Info   ] <Thu Nov 12 11:21:08 GMT +0000 2020> Advertise:true | Snapshot type:2 | Table
name:global.system.confrtbrickig.table | Table type:system_config_table | Deferred:false | Interval:0 |
Type:0 | Consume:false
[   Info   ] <Thu Nov 12 11:21:08 GMT +0000 2020> No keys to inherit, yang node identifier: table-type
system_config_table,  table-getter symbol name : confd_system_config_tbl_tmpl_get , libname : libconfd.so
[   Info   ] <Thu Nov 12 11:21:08 GMT +0000 2020> Advertise:true | Snapshot type:2 | Table
name:global.rtbrick.hostname.config | Table type:global_rtbrick_hostname_tbl | Deferred:false | Interval:0 |
Type:0 | Consume:false
[   Info   ] <Thu Nov 12 11:21:08 GMT +0000 2020> No keys to inherit, yang node identifier: table-type
global_rtbrick_hostname_tbl,  table-getter symbol name : confd_rtbrick_hostname_config_tbl_tmpl_get , libname
: libconfd.so
[   Info   ] <Thu Nov 12 11:21:08 GMT +0000 2020> Commit Success
```

Example 4: Show Log to File

```
supervisor@rtbrick: op> show log to file test.log
supervisor@rtbrick: op> exit
supervisor@rtbrick:~ $ cat test.log
[   info   ] <2022-05-10T11:53:15.399613+0000> Global config for Instance(default) is added
[   info   ] <2022-05-10T11:53:15.400666+0000> Global address family(ipv4, unicast) is added in
Instance(default)
[   info   ] <2022-05-10T11:53:15.400711+0000> Global address family(ipv4, multicast) is added in
Instance(default)
[   info   ] <2022-05-10T11:53:15.400729+0000> Global address family(ipv4, labeled-unicast) is added in
Instance(default)
[   info   ] <2022-05-10T11:53:15.400744+0000> Global address family(ipv6, unicast) is added in
Instance(default)
[   info   ] <2022-05-10T11:53:15.400758+0000> Global address family(ipv6, multicast) is added in
Instance(default)
[   info   ] <2022-05-10T11:53:15.400773+0000> Global address family(ipv6, labeled-unicast) is added in
Instance(default)
[   info   ] <2022-05-10T11:53:15.400787+0000> Global address family(mpls, unicast) is added in
Instance(default)
[   info   ] <2022-05-10T11:53:07.757687+0000> User 'supervisor' executed command 'show log'
[   info   ] <2022-05-10T11:53:10.751083+0000> User 'supervisor' executed command 'show log'
[   info   ] <2022-05-10T11:53:15.338391+0000> Failed due to bds events- {table:global_ntp_config_tbl}
[   info   ] <2022-05-10T11:53:15.338442+0000> Failed due to bds events- {table:ipmi_user_config_table}
[   info   ] <2022-05-10T11:53:15.338469+0000> Failed due to bds events- {table:lum_config_table}
[   info   ] <2022-05-10T11:53:15.338491+0000> Failed due to bds events- {table:ipmi_interface_config_table}
[   info   ] <2022-05-10T11:53:15.338509+0000> Failed due to bds events- {table:authorization_config_table}
[   info   ] <2022-05-10T11:53:07.644676+0000> Commit Success
[   info   ] <2022-05-10T11:53:15.337573+0000> Table - global_tacacs_config_tbl object not found event
secure_hostconfd_write_config
[   info   ] <2022-05-10T11:53:15.337602+0000> Table [global.tacacs.config] Object [name -
tacacs_config_object] attribute - tacacs_server_ip not found event TACACS Server Hostconfd Config
```

```
[   info   ] <2022-05-10T11:53:15.337241+0000> No objects present in alert configuration table to send to
hostconfd
```

Example 5: Logs for filter level

```
supervisor@rtbrick: op> show log filter level Error
[  Error   ] <Tue Nov 10 19:44:31 GMT +0000 2020> Table
[/var/rtbrick/commit_registry/global.commit.registry.snap] - event Could not open file for reading
[  Error   ] <Tue Nov 10 19:44:48 GMT +0000 2020> Table [global.tacacs.config] Object [name -
tacacs_config_object] attribute - tacacs_server_ip not found event TACACS Server Hostconfd Config
```

Example 6: Logs for specified module

```
supervisor@rtbrick: op> show log filter module secure_management
[  Error   ] <Tue Nov 10 19:44:48 GMT +0000 2020> Table [global.tacacs.config] Object [name -
tacacs_config_object] attribute - tacacs_server_ip not found event TACACS Server Hostconfd Config
supervisor@leaf: op>
```

Example 7: View of the logs in abstract format

```
supervisor@rtbrick: op> show log format abstract
Table [/var/rtbrick/commit_registry/global.commit.registry.snap] - event Could not open file for reading
Commit Success
CLI candidate config deletion begin
CLI candidate config deletion ends, status : success
CLI candidate config addition begin
Advertise:true | Snapshot type:2 | Table name:global.system.config.table | Table type:system_config_table |
Deferred:false | Interval:0 | Type:0 | Consume:false
No keys to inherit, yang node identifier: table-type system_config_table,  table-getter symbol name :
confd_system_config_tbl_tmpl_get , libname : libconfd.so
Setting attribute > Table name : global.system.config.table, object : system_config_object, command-token-
name : name, attribute-name : configuration_name, value : rtbrick, type : string
BDS object found
Processing TARGET transaction and replaying ADD, xml_name : system
Setting attribute > Table name : global.system.config.table, object : system_config_object, command-token-
name : name, attribute-name : configuration_name, value : rtbrick, type : string
Table name global.system.config.table, object name system_config_object
Table name global.system.config.table, object name system_config_object, status success
Advertise:true | Snapshot type:2 | Table name:global.rtbrick.hostname.config | Table
type:global_rtbrick_hostname_tbl | Deferred:false | Interval:0 | Type:0 | Consume:false
No keys to inherit, yang node identifier: table-type global_rtbrick_hostname_tbl,  table-getter symbol name :
confd_rtbrick_hostname_config_tbl_tmpl_get , libname : libconfd.so
```

BDS Logging Clear Commands

Clear commands allow you to delete existing logs.

Clear Logs

This commands resets all logs.

Syntax:

clear log <option> ...

Option	Description
bd <bd-name>	Clear all BDS logs from the given BD.
table <table-name>	Clears the specified log table.

Viewing ONL Log Files

ONL log files are available at the following directory:

CtrlD log files are available at:

/var/log/rtbrick-ctrld.log`

Example: CtrlD Logs

```
supervisor@onl>ufi06.q2c.u25.r4.nbg.rtbrick.net:/var/log $ tail -10 rtbrick-
ctrld.log
2022-05-04 08:31:24 UTC INF HTTP request completed host=198.51.100.29:19091
method=GET
path=/api/v1/rbfs/elements/ufi06.q2c.u25.r4.nbg.rtbrick.net/services/prometheus/pr
oxy/federate remote_addr=198.51.100.49:41508 request_id=R4n9-tadb statuscode=200
user_name= user_subject=
2022-05-04 08:31:24 UTC INF HTTP request completed host=198.51.100.29:19091
method=GET path=/api/v1/ctrld/info remote_addr=198.51.100.121:34158
request_id=yFn9Ptadb statuscode=200 user_name= user_subject=
2022-05-04 08:31:24 UTC INF HTTP request completed host=198.51.100.31:19091
method=GET path=/api/v1/ctrld/system/clock remote_addr=198.51.100.10:57824
request_id=_cno-gadu statuscode=200 user_name= user_subject=
2022-05-04 08:31:25 UTC INF HTTP request completed host=198.51.100.29:19091
method=GET path=/api/v1/rbfs/elements/rtbrick/services/PROMETHEUS/proxy/federate
remote_addr=198.51.100.229:39654 request_id=DXSoPgaHu statuscode=404 user_name=
user_subject=
2022-05-04 08:31:29 UTC INF HTTP request completed host=198.51.100.29:19091
method=GET
path=/api/v1/rbfs/elements/ufi06.q2c.u25.r4.nbg.rtbrick.net/services/prometheus/pr
oxy/federate remote_addr=198.51.100.49:41508 request_id=iQqo-gaHu statuscode=200
user_name= user_subject=
2022-05-04 08:31:34 UTC INF HTTP request completed host=198.51.100.29:19091
method=GET
path=/api/v1/rbfs/elements/ufi06.q2c.u25.r4.nbg.rtbrick.net/services/prometheus/pr
oxy/federate remote_addr=198.51.100.49:41508 request_id=LWM9PgrHb statuscode=200
user_name= user_subject=
```

ApiGwD log files are available at:

/var/log/rtbrick-apigwd.log

Example: API Gateway Logs

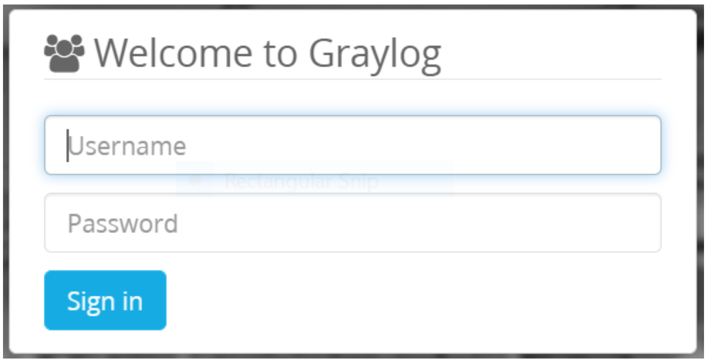
```
supervisor@onl>ufi06.q2c.u25.r4.nbg.rtbrick.net:/var/log $ cat rtbrick-apigwd.log
```

```
Tue May  3 23:27:12 UTC 2022 Starting rtbrick apigwd service
Version: v0.10.0-internal.20220222110916+Bdevelopment.C2a896336 (built with
go1.17.7)
2022-05-03 23:27:12 UTC INF development/apigwd/pkg/options/options.go:158 >
watching for file change /etc/rtbrick/apigwd/config.json
2022-05-03 23:27:12 UTC INF development/apigwd/cmd/apigwd/server.go:29 > listening
on listen_addr=:12321
2022-05-03 23:27:12 UTC INF development/apigwd/cmd/apigwd/server.go:89 > certman:
certificate and key loaded
2022-05-03 23:27:12 UTC INF development/apigwd/cmd/apigwd/server.go:89 > certman:
watching for cert and key change
2022-05-03 23:28:15 UTC INF development/apigwd/pkg/options/options.go:165 > watch
event: {/etc/rtbrick/apigwd/config.json 2}
2022-05-03 23:28:15 UTC INF development/apigwd/pkg/options/options.go:165 > watch
event: {/etc/rtbrick/apigwd/config.json 2}
2022-05-03 23:28:15 UTC INF development/apigwd/cmd/apigwd/routes.go:62 > reloaded
request limiter config
2022-05-03 23:28:15 UTC INF development/apigwd/cmd/apigwd/routes.go:62 > reloaded
request limiter config
```

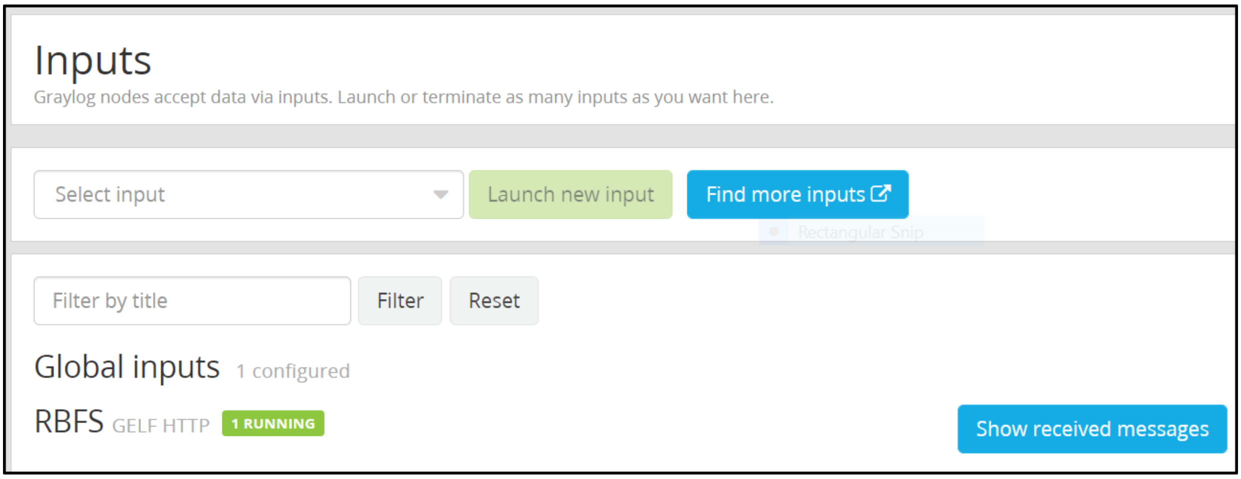
Viewing Logs in Graylog

For viewing your log data on Graylog, perform the following steps:

1. Open the Graylog webpage.
2. Log in using your user credentials.



3. Click **System** and select **Input**.



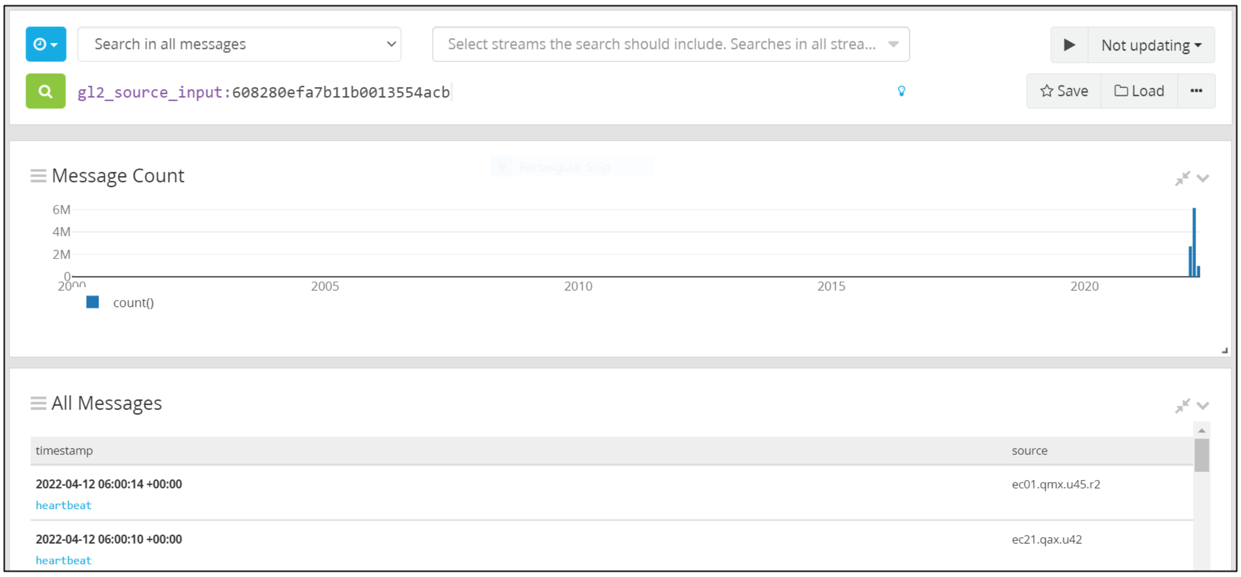
Inputs

Graylog nodes accept data via inputs. Launch or terminate as many inputs as you want here.

Global inputs 1 configured

RBFS GELF HTTP 1 RUNNING

4. Click the **Show received message** tab.



gl2_source_input:608280efa7b11b0013554acb

timestamp	source
2022-04-12 06:00:14 +00:00 heartbeat	ec01.qmx.u45.r2
2022-04-12 06:00:10 +00:00 heartbeat	ec21.qax.u42

The log messages page appears.

15.2.4. RBFS Log Reference

RBFS generates logs for various events. RBFS logs are record of events which occur on different functional components or modules in RBFS.

RBFS logs are categorized based on demons or modules from which they are produced. The following are some of the example scenarios when event logs are generated:

- Failure conditions such as failure in decoding received DHCP relay packet or failure to access a configuration file.
- Normal operations such as when an IPoE subscriber is created or deleted and an interface has been configured and so on.

- Critical conditions such as power failure.

This document provides log information such as log ID, message, description and module, severity level and recommended actions.

Log Destination You can locate and view the logs using multiple interfaces which include RBFS command line interface. RBFS also supports exporting logs to external log management servers such as Syslog server and Graylog server for real-time analysis.

You can view the logs in the RBFS command line interface using the show log command. Various command options are available to filter and display logs. For more information, see the Logging Operational Commands section.

BD Log Messages

Log ID	HTTP_SEND
Message	Send response for request endpoint %s %s buf count %s state %s
Description	Sent response details.
Log Module	bd
Log Group	http
Severity	INFO
Recommended Actions	No action required.

Log ID	HTTP_JWK_FILE_READ_FAILED
Message	Failed to read keys from JWK file %s
Description	Failed to read keys from the JSON Web Key (JWK) file.
Log Module	bd
Log Group	http
Severity	ERROR
Recommended Actions	Contact RtBrick Customer Support.

Log ID	HTTP_JWK_MULTIPLE_KEYS
Message	Multiple keys in JWK secret file
Description	Multiple keys in the JSON Web Key secret file.
Log Module	bd
Log Group	http
Severity	ERROR
Recommended Actions	Contact RtBrick Customer Support

Log ID	HTTP_JWK_FILE_MISSING
Message	JSON Web Key file %s not found
Description	The JSON Web Key file is not available.
Log Module	bd
Log Group	http
Severity	ERROR
Recommended Actions	Contact RtBrick Customer Support.

Log ID	HTTP_JWK_FILE_MEM_ALLOC_FAILED
Message	Memory error: unable to allocate %s bytes
Description	Unable to allocate memory to read the JSON Web Key file.
Log Module	bd
Log Group	http
Severity	ERROR
Recommended Actions	Contact RtBrick Customer Support.

Log ID	HTTP_JWK_FILE_OPEN_FAILED
Message	Unable to open JWK file %s
Description	Unable to open the JSON Web Key file.
Log Module	bd

Log Group	http
Severity	ERROR
Recommended Actions	Contact RtBrick customer support.

Log ID	HTTP_JWK_MISSING_KEY
Message	Key missing in JWK secret file
Description	Key is missing in the JSON web key secret file.
Log Module	bd
Log Group	http
Severity	ERROR
Recommended Actions	Contact RtBrick customer support.

Log ID	HTTP_JWK_FILE_JSON_PARSE_FAILED
Message	Failed to parse json content from JWK file %s
Description	Failed to parse the JSON content from the JWK file.
Log Module	bd
Log Group	http
Severity	ERROR
Recommended Actions	Contact RtBrick customer support.

Log ID	HTTP_WRITE_BUFFER_MEM_ALLOC_FAILED
Message	Failed to allocate memory for HTTP write buffer of size %s
Description	Failed to allocate memory for the HTTP write buffer.
Log Module	bd
Log Group	http
Severity	ERROR
Recommended Actions	Contact RtBrick customer support.

Log ID	HTTP_WRITE_PRINTF_FAILED
Message	Failed to write data to http write buffer
Description	Failed to write data to the HTTP write buffer.
Log Module	bd
Log Group	http
Severity	ERROR
Recommended Actions	Contact RtBrick customer support.

BGP Log Messages

Log ID	BGP_PEER_STATE_CHANGE
Message	BGP FSM change, peer %s, source %s, hostname %s, instance %s changed state from %s to %s, reason %s
Description	BGP FSM state has been changed.
Log Module	bgp
Log Group	peer
Severity	Alert
Recommended Actions	If the session is down, verify the reachability of the peer router.

Log ID	BGP_IFL_NBR_ADD
Message	Link-local peer %s discovered on interface %s, instance %s
Description	BGP link-local peer is discovered on the interface.
Log Module	bgp
Log Group	peer
Severity	Notice
Recommended Actions	No action required

Log ID	BGP_IFL_NBR_DEL
Message	Link-local peer %s deleted on interface %s, instance %s

Description	BGP link-local peer has been deleted on the interface.
Log Module	bgp
Log Group	peer
Severity	Notice
Recommended Actions	No action required

Log ID	BGP_PEER_RESET
Message	BGP peer %s, source %s, hostname %s, instance %s reset, reason %s
Description	BGP session reset occurred.
Log Module	bgp
Log Group	peer
Severity	Notice
Recommended Actions	If the session is down, verify the reachability of the peer.

Log ID	BGP_PEER_RESET_FSM_ERR
Message	BGP peer %s, source %s, hostname %s, instance %s reset, reason FSM error subcode %s, message type %s
Description	BGP session reset occurred due to an FSM error.
Log Module	bgp
Log Group	peer
Severity	Info
Recommended Actions	No action required

Log ID	BGP_MSG_OPEN_VERSION
Message	BGP peer %s, source %s, hostname %s, instance %s received open message with version %s
Description	Recieved a BGP Open message with an unsupported version number. Only version 4 is supported.

Log Module	bgp
Log Group	peer
Severity	Info
Recommended Actions	Check whether the other BGP peer supports the BGP version 4.

Log ID	BGP_MSG_OPEN_IDENTIFIER
Message	BGP Open message received with bad identifier %s
Description	BGP Open message is received with a bad identifier.
Log Module	bgp
Log Group	peer
Severity	Info
Recommended Actions	Peer router ID and local router ID are configured with the same value, update the configuration correctly.

Log ID	BGP_MSG_OPEN_HOLDTIMER
Message	BGP peer %s, source %s, hostname %s, instance %s received open message with illegal hold timer %s
Description	BGP Open message is received with a bad hold timer value.
Log Module	bgp
Log Group	peer
Severity	Info
Recommended Actions	Configure the remote router with a holdtimer value more than ‘3’.

Log ID	BGP_NO_ROUTER_ID
Message	Router ID is not present in instance %s
Description	BGP Router ID is not present.
Log Module	bgp
Log Group	peer
Severity	Info

Recommended Actions	Configure the local router with the router ID.

Log ID	BGP_NO_LOCAL_AS
Message	Local AS is not configured in instance %s
Description	BGP ‘local-as’ parameter is not defined in the configuration.
Log Module	bgp
Log Group	peer
Severity	Info
Recommended Actions	Configure the BGP local AS in the BGP peer group.

Log ID	BGP_INVALID_RECEIVED_AS
Message	BGP peer %s, source %s, hostname %s, instance %s received wrong AS %s, expected AS %s
Description	BGP received a wrong AS number from a remote peer.
Log Module	bgp
Log Group	peer
Severity	Info
Recommended Actions	Verify that the ‘local-as’ parameter is defined in the remote BGP speaker and the ‘remote-as’ parameter is defined in the peer group configured in the local node are the same.

Log ID	BGP_MSG_MANDATORY_ATTRIBUTE
Message	BGP peer %s, source %s, hostname %s, instance %s received message %s with mandatory parameter %s missing
Description	Mandatory parameter for BGP message is not present.
Log Module	bgp
Log Group	peer
Severity	Info
Recommended Actions	Contact RtBrick customer support with the packet captured from the peer node.

Log ID	BGP_PEER_RESET_MSG_HDR_LEN_ERR
Message	BGP peer %s, source %s, hostname %s, instance %s is reset, reason: invalid message header length %s
Description	BGP peer is reset due to invalid header length.
Log Module	bgp
Log Group	peer
Severity	Info
Recommended Actions	Contact RtBrick customer support with the packet scenario captured from the peer node.

Log ID	BGP_PEER_RESET_INVALID_TYPE
Message	BGP peer %s, source %s, hostname %s, instance %s is reset, reason: invalid message type %s, length %s
Description	BGP peer is reset due to invalid message type.
Log Module	bgp
Log Group	peer
Severity	Info
Recommended Actions	BGP does not support the corresponding feature (message).

Log ID	BGP_PEER_RESET_INVALID_PARAMETER_TYPE
Message	BGP peer %s, source %s, hostname %s, instance %s is reset, reason: invalid optional parameter, message %u, parameter type %u
Description	BGP peer is reset due to invalid optional parameter type.
Log Module	bgp
Log Group	peer
Severity	Info
Recommended Actions	BGP does not support the corresponding feature (message).

Log ID	BGP_PEER_RESET_INVALID_CAP_LEN

Message	BGP peer %s, source %s hostname %s, instance %s is reset, reason: invalid capability type %s, capability length %s
Description	BGP peer is reset due to invalid capability length.
Log Module	bgp
Log Group	peer
Severity	Info
Recommended Actions	Contact customer support with the packet scenario captured from the peer node.

Log ID	BGP_DUPLICATE_TCP_ACCEPT
Message	BGP peer %s, source %s, hostname %s, instance %s duplicate accept socket is success.
Description	In this error scenario when one side BGP peer is up and on the other part TCP connection goes down and comes up again.
Log Module	bgp
Log Group	peer
Severity	Info
No action required	Contact Rtbrick customer support

Log ID	BGP_IFL_CHANGE
Message	Interface %s, instance %s received event %s
Description	Interface status has changed.
Log Module	bgp
Log Group	interface
Severity	Info
Recommended Actions	Verify the physical status of the interface.

Log ID	BGP_IFA_CHANGE
Message	Interface %s, instance %s received %s of address %s

Description	Event related to interface address.
Log Module	bgp
Log Group	interface
Severity	Info
Recommended Actions	Verify the physical status of the interface.

Log ID	BGP_INSTANCE_INACTIVE
Message	Instance %s is inactive, reason %s
Description	BGP instance became inactive.
Log Module	bgp
Log Group	instance
Severity	Info
Reco	No action required

Log ID	BGP_MSG_VALIDATION_FAIL
Message	BGP message received from peer %s, source %s, hostname %s, instance %s validation failed. Message type %s
Description	BGP Message validation failed.
Log Module	bgp
Log Group	message
Severity	Info
Recommended Actions	Contact customer support with the packet captured from the peer node.

Log ID	BGP_MSG_DECODE_FAIL
Message	BGP message received from peer %s, source %s, hostname %s, instance %s decode failed. Message type %s
Description	BGP Message decoding has failed.
Log Module	bgp
Log Group	message

Severity	Info
Recommended Actions	Contact customer support with the packet captured from the peer node.

Log ID	BGP_UPD_SET_ERR_INFO
Message	Update processing error. code %s , subcode %s
Description	BGP Update processed error code and subcodes.
Log Module	bgp
Log Group	message
Severity	Info
Recommended Actions	Verify that the feature is supported.

Log ID	BGP_UPD_INVALID_IPV4_PREFIX
Message	BGP update message processing, peer %s, source %s, hostname %s, instance %s: Received invalid prefix %s/%s
Description	BGP Update message is received with an invalid prefix.
Log Module	bgp
Log Group	message
Severity	Info
Recommended Actions	Capture the packet scenario and verify it.

Log ID	BGP_UPD_ATTR_REPEAT
Message	BGP update message processing, peer %s, source %s, hostname %s, instance %s: Attribute %s repeated
Description	BGP Update message is processed with a repeated attribute.
Log Module	bgp
Log Group	message
Severity	Info
Recommended Actions	Verify the neighbor router configurations.

Log ID	BGP_UPD_ATTR_WELLKNOWN_TRANS
Message	BGP update message processing, peer %s, source %s, hostname %s, instance %s: Attribute %s does not have well known transitive flag set
Description	BGP Update process error. Attribute does not have a well known transitive flag set.
Log Module	bgp
Log Group	message
Severity	Info
Recommended Actions	Verify the neighbor router configuration.

Log ID	BGP_UPD_ATTR_WK_TRANS_PARTIAL
Message	BGP update message processing, peer %s, source %s, hostname %s, instance %s: Well known transitive attribute %s has partial bit set
Description	BGP Update processing error: Well known transitive attribute has partial bit set
Log Module	bgp
Log Group	message
Severity	Info
Recommended Actions	Verify the neighbor router configurations.

Log ID	BGP_UPD_ATTR_INVALID_ASPATH_LEN
Message	BGP update message processing, peer %s, source %s, hostname %s, instance %s: Invalid AS path length %s
Description	BGP Update processed Invalid AS path length
Log Module	bgp
Log Group	message
Severity	Info

Recommended Actions	Verify that only 4 byte AS is enabled on peer BGP speakers. RBFS supports only 4 byte AS.

Log ID	BGP_UPD_ENFORCE_FIRST_AS
Message	BGP update message processing, peer %s, source %s, hostname %s, instance %s: First AS %s is not peer AS %s, ignoring
Description	BGP Update processed: ‘First AS is not peer AS, so ignore the update’ condition.
Log Module	bgp
Log Group	message
Severity	Info
Recommended Actions	Disable 'enforce first AS' attribute, if this is required.

Log ID	BGP_UPD_INVALID_SEG_TYPE
Message	BGP update message processing, peer %s, source %s, hostname %s, instance %s: Invalid path segment type %s
Description	BGP Update processed ‘invalid segment type’ condition.
Log Module	bgp
Log Group	message
Severity	Info
Recommended Actions	Check whether that the feature is supported.

Log ID	BGP_UPD_INVALID_NH
Message	BGP update message processing, peer %s, source %s, hostname %s, instance %s: Invalid unicast ip address in nexthop %s
Description	BGP Update processing: Invalid unicast ip address in nexthop
Log Module	bgp
Log Group	message
Severity	Info

Recommended Actions	Contact Rtbrick customer support

Log ID	BGP_UPD_INVALID_IP6_PREFIX
Message	BGP update message processing, peer %s, source %s, hostname %s, instance %s: Invalid prefix %s/%s
Description	BGP Update processed an invalid IPv6 prefix which is received from the peer.
Log Module	bgp
Log Group	message
Severity	Info
Recommended Actions	Capture and verify the packet scenario.

Log ID	BGP_UPD_RCVD_NON_NEGOTIATED_AF
Message	BGP update message processing, peer %s, source %s, hostname %s, instance %s: Received NLRI of non-negotiated address family %s, safi %s
Description	BGP Update processed a non-negotiated address family received from the peer (‘Received NLRI of non-negotiated address family’)
Log Module	bgp
Log Group	message
Severity	Info
Recommended Actions	Disable the address family on the peer node.

Log ID	BGP_UPD_INVALID_LEN
Message	BGP update message processing, peer %s, source %s, hostname %s, instance %s: Received invalid length %s
Description	BGP Update processed an invalid length which has been received from the peer.
Log Module	bgp

Log Group	message
Severity	Info
Recommended Actions	Check whether the feature is supported.

Log ID	BGP_UPD_MALFORMED_PATH_ATTR
Message	BGP update message processing, peer %s, source %s, hostname %s, instance %s: Received malformed path attribute %s, length %s
Description	BGP Update processes a malformed path attribute which has been received.
Log Module	bgp
Log Group	message
Severity	Info
Recommended Actions	Check whether the feature is supported.

Log ID	BGP_UPD_ATTR_EXT_LEN
Message	BGP update message processing, peer %s, source %s, hostname %s, instance %s: Invalid length bit set for attribute %s
Description	BGP Update processed an ‘Invalid length bit set for attribute’.
Log Module	bgp
Log Group	message
Severity	Info
Recommended Actions	Check whether the feature is supported.

Log ID	BGP_UPD_ATTR_INVALID_ORIGIN
Message	BGP update message processing, peer %s, source %s, hostname %s, instance %s: Invalid origin attribute value %s
Description	BGP Update processed an invalid ‘origin’ attribute value.
Log Module	bgp

Log Group	message
Severity	Info
Recommended Actions	Verify the neighbor router configuration.

Log ID	BGP_UPD_ATTR_OPTIONAL_TRANS
Message	BGP update message processing, peer %s, source %s, hostname %s, instance %s: Attribute %s does not have optional transitive flag set
Description	BGP Update processed attributes which do not have an optional transitive flag set.
Log Module	bgp
Log Group	message
Severity	Info
Recommended Actions	Verify the neighbor router configuration.

Log ID	BGP_UPD_OWN_ROUTER_ID
Message	BGP update message processing, peer %s, source %s, hostname %s, instance %s: Received own router id %s
Description	BGP Update processed its own router ID which has been received.
Log Module	bgp
Log Group	message
Severity	Info
Recommended Actions	Verify whether the same router ID is configured on both of the BGP speakers.

Log ID	BGP_UPD_ATTR_OPTIONAL_NON_TRANS
Message	BGP update message processing, peer %s, source %s, hostname %s, instance %s: Attribute %s does not have optional non-transitive flag set

Description	BGP Update processed attribute that does not have an optional non-transitive flag set.
Log Module	bgp
Log Group	message
Severity	Info
Recommended Actions	Verify the neighbor router configuration.

Log ID	BGP_UPD_INVALID_ATTR_EBGP
Message	BGP update message processing, peer %s, source %s, hostname %s, instance %s: Received invalid attribute %s from EBGP peer
Description	BGP Update processed an invalid attribute received from eBGP peer router.
Log Module	bgp
Log Group	message
Severity	Info
Recommended Actions	Verify the neighbor router configuration.

Log ID	BGP_UPD_INVALID_SR_IDX_LEN
Message	BGP update message processing, peer %s, source %s, hostname %s, instance %s: Invalid SR index length %s
Description	BGP Update processed an invalid SR index length.
Log Module	bgp
Log Group	message
Severity	Info
Recommended Actions	Verify the neighbor router configuration.

Log ID	BGP_UPD_INVALID_SRGB_LEN
Message	BGP update message processing, peer %s, source %s, hostname %s, instance %s: Invalid SRGB length %s

Description	BGP Update processed an invalid SRGB length.
Log Module	bgp
Log Group	message
Severity	Info
Recommended Actions	Verify the neighbor router configuration.

Log ID	BGP_UPD_PATH_ATTR_FAIL
Message	BGP update message processing, peer %s, source %s, hostname %s, instance %s: Processing path attribute failed
Description	BGP Update processed a failed path attribute.
Log Module	bgp
Log Group	message
Severity	Info
Recommended Actions	This can be a packet decoding problem. Send the captured packet to the Contact RtBrick customer support team for verification.

Log ID	BGP_UPD_MREACH_FAIL
Message	BGP update message processing, peer %s, source %s, hostname %s, instance %s: Processing MPREACH NLRI attribute failed
Description	BGP Update processed a failed MPREACH NLRI attribute.
Log Module	bgp
Log Group	message
Severity	Info
Recommended Actions	This can be a packet decoding problem. Send the captured packet to the Contact RtBrick customer support team for verification.

Log ID	BGP_UPD_INVALID_EXP_LEN

Message	BGP update message processing, peer %s, source %s, hostname %s, instance %s: Received invalid length %s . Expected %s
Description	BGP Update processed an invalid length which has been received.
Log Module	bgp
Log Group	message
Severity	Info
Recommended Actions	This can be a packet decoding problem. Send the captured packet to the Contact RtBrick customer support team for verification.

Log ID	BGP_UPD_NLRI_FAIL
Message	BGP update message processing, peer %s, source %s, hostname %s, instance %s: Processing NLRI attribute failed
Description	BGP Update processed a failed NLRI attribute.
Log Module	bgp
Log Group	message
Severity	Info
Recommended Actions	This can be a packet decoding problem. Send the captured packet scenario to the Contact RtBrick customer support team for verification.

Log ID	BGP_FSM_PEER_STATE_CHANGE
Message	Peer [%s] Src[%s] CurState[%s] Event[%s] NextState[%s]
Description	BGP Update processed a failed NLRI attribute.
Log Module	bgp
Log Group	message
Severity	Info
Recommended Actions	This can be a packet decoding problem. Send the captured packet scenario to the Contact RtBrick customer support team for verification.

HOSTCONFD Log Messages

Log ID	HOSTCONFD_BDS_GENERIC_FAIL
Message	Failed due to bds events- {table:%s}
Description	Failed due to a BDS event.
Log Module	hostconfd
Log Group	bds
Severity	INFO
Recommended Actions	Verify the table or object in the BDS commands.

Log ID	HOSTCONFD_TABLE_CONFIG_READ_FAIL
Message	Failed to read static hostconfd table configuration- {file:%s}
Description	Failed to read static hostconfd table configuration.
Log Module	hostconfd
Log Group	config
Severity	INFO
Recommended Actions	Verify the json format/table-name in the configuration file.

Log ID	HOSTCONFD_DYN_PLUGIN_REGN_FAIL
Message	Failed to register dynamic plugin- {table-type:%s}
Description	Failed to register dynamic plugin.
Log Module	hostconfd
Log Group	config
Severity	INFO
Recommended Actions	Verify the table definition.

Log ID	HOSTCONFD_HTTP_HEADER_FAIL
Message	Failed to create HTTP header to send to hostconfd {event:%s}

Description	Failed to create HTTP header to send to the hostconfd daemon.
Log Module	hostconfd
Log Group	functional
Severity	INFO
Recommended Actions	Verify the table definition.

Log ID	HOSTCONFD_HTTP_CONTENT_FAIL
Message	Failed to create HTTP content to send to hostconfd- {table-type:%s}
Description	Failed to create HTTP content to send to hostconfd daemon.
Log Module	hostconfd
Log Group	functional
Severity	INFO
Recommended Actions	Verify the table definition.

Log ID	HOSTCONFD_CONNECTION_DOWN
Message	HOSTCONFD connection is DOWN {event:%s}
Description	HOSTCONFD connection is down.
Log Module	hostconfd
Log Group	functional
Severity	INFO
Recommended Actions	Verify the hostconfd operational status.

Log ID	HOSTCONFD_ALERTING_CONFIG_ERROR
Message	HOSTCONFD alert config error {event:%s}
Description	Error generated by hostconfd on alerting about a configuration error.
Log Module	hostconfd

Log Group	functional
Severity	ERROR
Recommended Actions	Verify the hostconfd operational status.

Log ID	IPMI_CONFIG_TABLE_ERROR
Message	Failed to create IPMI config table %s
Description	Failed to create the IPMI configuration table.
Log Module	hostconfd
Log Group	functional
Severity	ERROR
Recommended Actions	Contact RtBrick customer support.

Log ID	WATCHDOG_CONFIG_TABLE_ERROR
Message	Failed to create Watchdog config table %s
Description	Failed to create the watchdog configuration table.
Log Module	hostconfd
Log Group	functional
Severity	ERROR
Recommended Actions	Contact RtBrick Customer Support.

Interfaces Log Messages

Log ID	IFM_OPER_IFP_ADMIN_UP
Message	Admin status of the physical interface %s is UP
Description	The interface configuration has been changed.
Log Module	ifm
Log Group	config
Severity	info

Recommended Actions	Verify the interface configurations on the physical interface.

Log ID	IFM_OPER_IFP_ADMIN_DOWN
Message	Admin status of the physical interface %s is DOWN
Description	The interface configuration has been changed.
Log Module	ifm
Log Group	config
Severity	info
Recommended Actions	Verify the interface configurations for the physical interface.

Log ID	IFM_OPER_IFP_LINK_UP
Message	Link status of the physical interface %s is UP
Description	Physical link is up on the interface.
Log Module	ifm
Log Group	interface-events
Severity	info
Recommended Actions	Check the physical connectivity for this particular port.

Log ID	IFM_OPER_IFP_LINK_DOWN
Message	Link status of the physical interface %s is DOWN
Description	Physical link is down on the interface.
Log Module	ifm
Log Group	interface-events
Severity	info
Recommended Actions	Check the physical connectivity for this particular port.

Log ID	IFM_OPER_IFP_OPERATIONAL_UP
Message	Operational status of the physical interface %s is UP

Description	Admin-status or operational status of an interface has gone up.
Log Module	ifm
Log Group	interface-events
Severity	info
Recommended Actions	Check the physical connectivity for this particular port or verify the interface configuration.

Log ID	IFM_OPER_IFP_OPERATIONAL_DOWN
Message	Operational status of the physical interface %s is DOWN
Description	Admin-status or operational status of an interface has gone down.
Log Module	ifm
Log Group	interface-events
Severity	info
Recommended Actions	Check the physical connection for this particular port or the interface configuration

Log ID	IFM_OPER_IFL_ADMIN_UP
Message	Admin status of the logical interface %s is UP
Description	Logical interface configuration has been changed.
Log Module	ifm
Log Group	config
Severity	info
Recommended Actions	Verify the interface configurations for this logical interface.

Log ID	IFM_OPER_IFL_ADMIN_DOWN
Message	Admin status of the logical interface %s is DOWN
Description	The logical interface configuration has been changed.
Log Module	ifm
Log Group	config

Severity	info
Recommended Actions	Check the interface configurations for this logical interface.

Log ID	IFM_OPER_IFL_LINK_UP
Message	Link status of the logical interface %s is UP
Description	Link is up on the logical interface.
Log Module	ifm
Log Group	interface-events
Severity	info
Recommended Actions	Check the connectivity for this particular logical interface.

Log ID	IFM_OPER_IFL_LINK_DOWN
Message	The link status of the logical interface %s is DOWN
Description	Link is down on the logical interface.
Log Module	ifm
Log Group	interface-events
Severity	info
Recommended Actions	Check the connectivity for this particular logical interface.

Log ID	IFM_OPER_IFL_OPERATIONAL_UP
Message	Operational status of the logical interface %s is UP
Description	Admin-status or operational status of a logical interface has gone up.
Log Module	ifm
Log Group	interface-events
Severity	info
Recommended Actions	Check the connectivity for this particular port or verify the logical interface configuration.

Log ID	IFM_OPER_IFL_OPERATIONAL_DOWN
Message	The operational status of the logical interface %s is DOWN
Description	Admin-status or operational status of a logical interface has gone down.
Log Module	ifm
Log Group	interface-events
Severity	info
Recommended Actions	Check the physical connectivity for this particular port or verify the logical interface configuration.

Log ID	IFM_OPER_IFL_IPV4_STATUS_UP
Message	IPv4 status of the logical interface %s is UP
Description	IPv4 service has been enabled on a logical interface.
Log Module	ifm
Log Group	config
Severity	info
Recommended Actions	Verify the IPv4 status for this logical interface.

Log ID	IFM_OPER_IFL_IPV4_STATUS_DOWN
Message	IPv4 status of the logical interface %s is DOWN
Description	Logical Interface configuration change triggered
Log Module	ifm
Log Group	config
Severity	info
Recommended Actions	Verify the IPv4 status for this logical interface.

Log ID	IFM_OPER_IFL_IPV6_STATUS_UP
Message	IPv6 status of the logical interface %s is UP
Description	IPv6 service is enabled on a logical interface.

Log Module	ifm
Log Group	config
Severity	info
Recommended Actions	Verify the IPv6 status for this logical interface.

Log ID	IFM_OPER_IFL_IPV6_STATUS_DOWN
Message	IPv6 status of the logical interface %s is DOWN
Description	IPv6 service is disabled on a logical interface.
Log Module	ifm
Log Group	config
Severity	info
Recommended Actions	Verify the IPv6 status for this logical interface.

Log ID	IFM_OPER_IFL_MPLS_STATUS_UP
Message	The MPLS status of the logical interface %s is UP
Description	MPLS service is enabled on a logical interface.
Log Module	ifm
Log Group	config
Severity	info
Recommended Actions	Verify the status of MPLS for this logical interface.

Log ID	IFM_OPER_IFL_MPLS_STATUS_DOWN
Message	MPLS status of the logical interface %s is DOWN
Description	MPLS service is disabled on a logical interface.
Log Module	ifm
Log Group	config
Severity	info

Recommended Actions	Verify the status of MPLS for this logical interface.

IPoE Log Messages

Log ID	SUBSCRIBER_CREATED
Message	Subscriber %s created for %s.%s:%s %s
Description	IPoE subscriber has been created.
Log Module	ipoe
Log Group	subscriber
Severity	info
Recommended Actions	No action required.

Log ID	SUBSCRIBER_DELETED
Message	Subscriber %s deleted with terminate code %s
Description	IPoE subscriber has been deleted.
Log Module	ipoe
Log Group	subscriber
Severity	info
Recommended Actions	No action required.

Log ID	DHCP_BINDING_CREATED
Message	Subscriber %s DHCP binding created
Description	Subscriber DHCP binding (an entry into the binding table) has been created.
Log Module	ipoe
Log Group	dhcp
Severity	info
Recommended Actions	No action required.

Log ID	DHCP_BINDING_DELETED
Message	Subscriber %s DHCP lease expired
Description	Subscriber DHCP binding has been deleted.
Log Module	ipoe
Log Group	dhcp
Severity	info
Recommended Actions	No action required.

Log ID	DHCP_BINDING_EXPIRED
Message	Subscriber %s DHCP binding lease timer expired
Description	Subscriber DHCP binding lease expired.
Log Module	ipoe
Log Group	dhcp
Severity	info
Recommended Actions	No action required.

Log ID	DHCPV6_BINDING_CREATED
Message	Subscriber %s DHCPv6 binding created
Description	Subscriber DHCPv6 binding has been created.
Log Module	ipoe
Log Group	dhcp
Severity	info
Recommended Actions	No action required.

Log ID	DHCPV6_BINDING_DELETED
Message	Subscriber %s DHCPv6 lease expired
Description	Subscriber DHCPv6 binding has been deleted.
Log Module	ipoe

Log Group	dhcp
Severity	info
Recommended Actions	No action required.

Log ID	DHCPV6_BINDING_EXPIRED
Message	Subscriber %s DHCPv6 binding lease timer expired
Description	The subscriber DHCPv6 binding lease expired.
Log Module	ipoe
Log Group	dhcp
Severity	info
Recommended Actions	No action required.

Log ID	SUBSCRIBER_DELETE_FAILED
Message	Subscriber %s deletion with terminate code %s failed
Description	IPoE subscriber deletion failure.
Log Module	ipoe
Log Group	subscriber
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSCRIBER_RD_FAILURE
Message	Subscriber %s from redundancy session %s failed due to %s
Description	IPoE redundancy subscriber failure.
Log Module	ipoe
Log Group	redundancy
Severity	debug
Recommended Actions	Contact RtBrick Customer Support.

Log ID	DHCP_RELAY_TRAP
Message	Trap rule %s %s for DHCP relay
Description	Event of DHCP relay trap installation.
Log Module	ipoe
Log Group	dhcp-relay
Severity	info
Recommended Actions	No action required.

Log ID	DHCP_RELAY_DECODE_ERROR
Message	DHCP relay packet decode error on interface %s, error code: %s subcode: %s
Description	Failure in decoding the received DHCP relay packet.
Log Module	ipoe
Log Group	dhcp-relay
Severity	debug
Recommended Actions	No action required.

Log ID	DHCP_RELAY_SHM_DECODE
Message	DHCP relay packet shm validation error
Description	Failure in decoding the received DHCP relay packet.
Log Module	ipoe
Log Group	dhcp-relay
Severity	info
Recommended Actions	No action required.

Log ID	DHCP_RELAY_IP_UDP_DECODE
Message	DHCP relay packet IP header decode error on interface %s
Description	Failure in decoding the received DHCP relay packet.

Log Module	ipoe
Log Group	dhcp-relay
Severity	debug
Recommended Actions	No action required.

Log ID	DHCP_RELAY_GATEWAY_IP
Message	DHCP relay gateway IP address not configured for interface %s
Description	DHCP relay configuration is not available.
Log Module	ipoe
Log Group	dhcp-relay
Severity	debug
Recommended Actions	Configure the DHCP relay gateway IP address

Log ID	DHCP_RELAY_BINDING_ADD
Message	DHCP relay binding added for client %s on interface %s with gateway IP %s, client-ip %s and lease-time %s
Description	DHCP relay binding has been added.
Log Module	ipoe
Log Group	dhcp-relay
Severity	debug
Recommended Actions	No action required.

Log ID	DHCP_RELAY_BINDING_DELETE
Message	DHCP relay binding deleted for client %s on interface %s
Description	DHCP relay binding is deleted.
Log Module	ipoe
Log Group	dhcp-relay
Severity	debug

Recommended Actions	No action required.

Log ID	DHCP_RELAY_BINDING_MISSING
Message	DHCP relay binding missing for client %s, ignoring the packet from server
Description	DHCP relay binding is not available.
Log Module	ipoe
Log Group	dhcp-relay
Severity	debug
Recommended Actions	Contact RtBrick Customer Support.

Log ID	DHCP_RELAY_CONFIG
Message	DHCP relay configuration missing for interface %s
Description	DHCP relay configuration is not available.
Log Module	ipoe
Log Group	dhcp-relay
Severity	debug
Recommended Actions	Configure DHCP relay.

Log ID	DHCP_RELAY_SERVER_LIST_CONFIG
Message	DHCP relay server configuration missing on interface %s, dropping the DHCP packet %s
Description	DHCP server configuration is not available.
Log Module	ipoe
Log Group	dhcp-relay
Severity	debug
Recommended Actions	Verify the DHCP server configuration.

Log ID	DHCP_RELAY_SERVER_CONFIG
Message	DHCP relay server IP configuration missing on interface %s for DHCP server %s
Description	DHCP server configuration is missing.
Log Module	ipoe
Log Group	dhcp-relay
Severity	debug
Recommended Actions	Verify the DHCP relay configuration.

Log ID	DHCP_RELAY_SERVER_GET
Message	Unable to get DHCP relay server %s configuration on interface %s
Description	DHCP server configuration is missing.
Log Module	ipoe
Log Group	dhcp-relay
Severity	debug
Recommended Actions	Configure DHCP server.

Log ID	DHCP_RELAY_PACKET_CLIENT_RCV
Message	DHCP %s received from client on interface %s with MAC %s
Description	DHCP relay packet received from the client.
Log Module	ipoe
Log Group	dhcp-relay
Severity	debug
Recommended Actions	No action required.

Log ID	DHCP_RELAY_PACKET_SERVER_RCV
Message	DHCP %s received from server %s for client on interface %s with MAC %s

Description	DHCP relay packet received from the server
Log Module	ipoe
Log Group	dhcp-relay
Severity	debug
Recommended Actions	No action required.

Log ID	DHCP_RELAY_PACKET_CLIENT_SENT
Message	DHCP %s from server %s sent to client on interface %s with MAC %s
Description	DHCP relay packet sent to the client.
Log Module	ipoe
Log Group	dhcp-relay
Severity	debug
Recommended Actions	No action required.

Log ID	DHCP_RELAY_PACKET_SERVER_SENT
Message	DHCP %s sent to server %s from client on interface %s with MAC %s
Description	DHCP relay packet sent to the server.
Log Module	ipoe
Log Group	dhcp-relay
Severity	debug
Recommended Actions	No action required.

Log ID	DHCP_RELAY_PACKET_CLIENT_RCV_INVALID
Message	Invalid DHCP %s received from client on interface %s with MAC %s
Description	DHCP relay packet received from the client.
Log Module	ipoe

Log Group	dhcp-relay
Severity	debug
Recommended Actions	No action required.

Log ID	DHCP_RELAY_PACKET_SERVER_RCV_INVALID
Message	Invalid DHCP %s received from server %s
Description	DHCP relay packet received from the server.
Log Module	ipoe
Log Group	dhcp-relay
Severity	debug
Recommended Actions	No action required.

Log ID	DHCP_RELAY_PROXY_SERVER
Message	Selecting the server %s for the client on interface %s in proxy mode
Description	DHCP relay proxy server has been selected.
Log Module	ipoe
Log Group	dhcp-relay
Severity	debug
Recommended Actions	No action required.

Log ID	DHCP_RELAY_PROXY_IGNORE_SERVER_PACKET
Message	Ignoring packet %s from server %s as the selected server is %s
Description	DHCP relay proxy packet ignored from the server.
Log Module	ipoe
Log Group	dhcp-relay
Severity	debug

Recommended Actions	No action required.

Log ID	DHCP_RELAY_ENCODE_ERROR
Message	Error in encoding DHCP %s packet with client interface %s with MAC %s
Description	DHCP relay encode error.
Log Module	ipoe
Log Group	dhcp-relay
Severity	debug
Recommended Actions	No action required.

Log ID	DHCP_RELAY_PROXY_NO_SERVER
Message	Server identifier missing in DHCP relay binding, dropping the DHCP %s packet from client interface %s with MAC %s
Description	No DHCP relay proxy server in binding.
Log Module	ipoe
Log Group	dhcp-relay
Severity	debug
Recommended Actions	No action required.

Log ID	DHCP_RELAY_INVALID_DST_IP
Message	Invalid destination IP, dropping the DHCP %s packet from client interface %s with MAC %s
Description	DHCP relay proxy server in binding is not available.
Log Module	ipoe
Log Group	dhcp-relay
Severity	debug
Recommended Actions	No action required.

IS-IS Log Messages

Log ID	ISIS_INSTANCE_OVERLOAD
Message	ISIS instance %s, level %s, %s, overload state
Description	ISIS instance overload enters or exits.
Log Module	isis
Log Group	instance
Severity	Alert
Recommended Actions	No action required.

Log ID	ISIS_NEIGHBOR_DISCOVERY
Message	ISIS neighbor %s, discovered, interface %s, instance %s
Description	An ISIS neighbor is discovered.
Log Module	isis
Log Group	neighbor
Severity	Alert
Recommended Actions	No action required.

Log ID	ISIS_NEIGHBOR_STATE_CHANGE
Message	ISIS neighbor %s, interface %s, instance %s, changed state from %s, to %s, reason %s
Description	Changes in the ISIS neighbor FSM state.
Log Module	isis
Log Group	neighbor
Severity	Alert
Recommended Actions	No action required.

Log ID	ISIS_NEIGHBOR_DELETED
Message	ISIS neighbor %s, deleted, interface %s, instance %s

Description	An ISIS neighbor is deleted.
Log Module	isis
Log Group	neighbor
Severity	Info
Recommended Actions	No action is required.

Log ID	ISIS_IFL_CHANGE
Message	Interface %s, instance %s, received event %s, operational state %s, admin state %s, ipv4 status %s, ipv6 status %s
Description	Event of ISIS enabled interface status change.
Log Module	isis
Log Group	interface
Severity	Info
Recommended Actions	No action required.

Log ID	ISIS_IFA_CHANGE
Message	Interface %s, instance %s, received %s, for address %s
Description	An ISIS-enabled interface address event.
Log Module	isis
Log Group	interface
Severity	Info
Recommended Actions	No action required.

Log ID	ISIS_PDU_DECODE_FAIL
Message	ISIS PDU %s, decode failure received in interface %s, instance %s, level %s, reason %s
Description	ISIS PDU decoding failure.
Log Module	isis
Log Group	message

Severity	Alert
Recommended Actions	Capture and verify the ISIS packets. Contact RtBrick Customer Support.

Log ID	ISIS_AUTHENTICATION_FAIL
Message	Authentication failure for PDU %s, interface %s, instance %s, level %s
Description	ISIS Authentication failure.
Log Module	isis
Log Group	message
Severity	Alert
Recommended Actions	Verify if the authentication type, key, and password are matching at both ends.

Log ID	ISIS_LSP_FLOOD_FILTER_BLOCK
Message	LSP flooding is blocked for neighbor %s, in interface %s, instance %s, level %s, for lsp-id %s
Description	Event of ISIS flood filter block.
Log Module	isis
Log Group	message
Severity	Debug
Recommended Actions	No action required.

Log ID	ISIS_SPF_COMPUTATION
Message	Dijkstra’s Shortest Path First algorithm triggered in instance %s, level %s. Reason %s
Description	The Dijkstra’s Shortest Path First algorithm triggered
Log Module	isis
Log Group	spf
Severity	Info

Recommended Actions	No action required.

Log ID	ISIS_SPF_COMPUTATION_LSP
Message	Dijkstra’s Shortest Path First algorithm triggered in instance %s, level %s. Reason LSP %s, received
Description	The received LSP has triggered Dijkstra’s Shortest Path First algorithm.
Log Module	isis
Log Group	spf
Severity	Info
Recommended Actions	No action required.

Log ID	ISIS_ORIGINATE_SELF_LSP
Message	Self lsp is originate in instance %s, level %s, lsp id %s, sequence number %s
Description	LSP originated.
Log Module	isis
Log Group	lsp
Severity	Info
Recommended Actions	No action is required.

Log ID	ISIS_ORIGINATE_LSP_PURGE
Message	Purge lsp is originated in instance %s, level %s, lsp id %s, sequence number %s
Description	Purge LSP originated.
Log Module	isis
Log Group	lsp
Severity	Info

Recommended Actions	No action required.

Log ID	ISIS_IPV4_PREFIX_ADD
Message	Add %s, metric %s, source %s, preference %s, instance %s, topology %s
Description	IPv4 prefix is added.
Log Module	isis
Log Group	route
Severity	Debug
Recommended Actions	No action required.

Log ID	ISIS_IPV6_PREFIX_ADD
Message	Add %s, metric %s, source %s, preference %s, instance %s, topology %s
Description	IPv6 prefix is added.
Log Module	isis
Log Group	route
Severity	Debug
Recommended Actions	No action required.

Log ID	ISIS_MPLS_LABEL_ADD
Message	Add mpls label %s, source %s, preference %s, instance %s, topology %s
Description	MPLS prefix is added.
Log Module	isis
Log Group	route
Severity	Debug
Recommended Actions	No action required.

Log ID	ISIS_IPV4_PREFIX_DELETE
Message	Delete %s, instance %s, topology %s
Description	IPv4 prefix is deleted.
Log Module	isis
Log Group	route
Severity	Debug
Recommended Actions	No action required.

Log ID	ISIS_IPV6_PREFIX_DELETE
Message	Delete %s, instance %s, topology %s
Description	IPv6 prefix is deleted.
Log Module	isis
Log Group	route
Severity	Debug
Recommended Actions	No action required.

Log ID	ISIS_MPLS_LABEL_DELETE
Message	Delete mpls-label %s, instance %s, topology %s
Description	MPLS prefix deletion log
Log Module	isis
Log Group	route
Severity	Debug
Recommended Actions	Contact Rtbrick customer support

LDP Log Messages

Log ID	LDP_SESSION_CREATED
Message	LDP Session created, peer %s, instance %s
Description	LDP has formed a new session.

Log Module	ldp
Log Group	peer
Severity	debug
Recommended Actions	None Required.

Log ID	LDP_SESSION_DELETED
Message	LDP Session deleted, peer %s, instance %s
Description	LDP session is deleted.
Log Module	ldp
Log Group	peer
Severity	debug
Recommended Actions	Run the show ldp session detail command and check the “Reason” attribute for information.

Log ID	LDP_TCP_STATE_CHNG
Message	LDP TCP state changed from "%s" to "%s" due to event "%s", peer %s, instance %s
Description	LDP TCP state has changed.
Log Module	ldp
Log Group	peer
Severity	debug
Recommended Actions	None Required.

Log ID	LDP_SESSION_STATE_CHNG
Message	LDP Session state changed from "%s" to "%s" due to event "%s", peer %s, instance %s
Description	LDP session state has changed.
Log Module	ldp
Log Group	peer
Severity	debug

Recommended Actions	None Required.

Log ID	LDP_MSG_ADDR_OUT
Message	LDP Address message sent for prefix %s, peer %s, instance %s
Description	LDP address message is sent.
Log Module	ldp
Log Group	peer
Severity	debug
Recommended Actions	None Required.

Log ID	LDP_MSG_ADDR_IN
Message	LDP Address message received for prefix %s, peer %s, instance %s
Description	LDP address message is received.
Log Module	ldp
Log Group	peer
Severity	debug
Recommended Actions	None Required.

Log ID	LDP_MSG_ADDR_WITHDRAW_OUT
Message	LDP Address withdraw message sent for prefix %s, peer %s, instance %s
Description	LDP address withdrawal message is sent.
Log Module	ldp
Log Group	peer
Severity	debug
Recommended Actions	None Required.

Log ID	LDP_MSG_ADDR_WITHDRAW_IN
Message	LDP Address withdraw message received for prefix %s, peer %s, instance %s
Description	LDP address withdrawal message is received.
Log Module	ldp
Log Group	peer
Severity	debug
Recommended Actions	None Required.

Log ID	LDP_MSG_LABEL_MAP_OUT
Message	LDP label mapping message sent for prefix %s with label-%s, peer %s, instance %s
Description	LDP mapping message is sent.
Log Module	ldp
Log Group	peer
Severity	debug
Recommended Actions	None Required.

Log ID	LDP_MSG_LABEL_MAP_IN
Message	LDP label mapping message received for prefix %s with label-%s, peer %s, instance %s
Description	LDP mapping is received.
Log Module	ldp
Log Group	peer
Severity	debug
Recommended Actions	None Required.

Log ID	LDP_MSG_LABEL_WITHDRAW_OUT

Message	LDP label withdraw message sent for prefix %s with label-%s, peer %s, instance %s
Description	LDP mapping withdrawal message is sent.
Log Module	ldp
Log Group	peer
Severity	debug
Recommended Actions	None Required.

Log ID	LDP_MSG_LABEL_WITHDRAW_IN
Message	LDP label withdraw message received for prefix %s with label-%s, peer %s, instance %s
Description	LDP mapping withdrawal message is received.
Log Module	ldp
Log Group	peer
Severity	debug
Recommended Actions	None Required.

Log ID	LDP_MSG_LABEL_REQ_REPLY_OUT
Message	LDP label request message reply for prefix %s sent with label-%s, peer %s, instance %s
Description	LDP request reply message is sent.
Log Module	ldp
Log Group	peer
Severity	debug
Recommended Actions	None Required.

Log ID	LDP_MSG_LABEL_REQ_IN
Message	LDP label request message received, prefix %s, peer %s, instance %s

Description	LDP request message is received.
Log Module	ldp
Log Group	peer
Severity	debug
Recommended Actions	None Required.

Log ID	LDP_MSG_LABEL_REL_OUT
Message	LDP label release message sent, prefix %s, label-%s, peer %s, instance %s
Description	LDP release message is sent.
Log Module	ldp
Log Group	peer
Severity	debug
Recommended Actions	None Required.

Log ID	LDP_MSG_LABEL_REL_IN
Message	LDP label release message received, prefix %s, label-%s, peer %s, instance %s
Description	LDP release message is received.
Log Module	ldp
Log Group	peer
Severity	debug
Recommended Actions	None Required.

Log ID	LDP_ADD_TO_FIB
Message	LDP route added to FIB, prefix %s, label-%s, instance %s
Description	LDP has been added to FIB.
Log Module	ldp
Log Group	peer

Severity	debug
Recommended Actions	None Required.

Log ID	LDP_DEL_FROM_FIB
Message	LDP route deleted from FIB, prefix %s, label-%s, instance %s
Description	LDP is deleted from FIB.
Log Module	ldp
Log Group	peer
Severity	debug
Recommended Actions	None Required.

Log ID	LDP_NBR_CREATED
Message	LDP Neighbour created, peer %s, interface %s, instance %s
Description	LDP has formed a new neighbor
Log Module	ldp
Log Group	adjacency
Severity	INFO
Recommended Actions	None Required.

Log ID	LDP_NBR_DELETED
Message	LDP Neighbour deleted, peer %s, interface %s, instance %s
Description	LDP neighbor is removed.
Log Module	ldp
Log Group	adjacency
Severity	INFO
Recommended Actions	Configure the neighbor router again for establishing neighborship and sending hello packets.

Log ID	LDP_NOTIFICATION_IN

Message	LDP Notification received, peer %s, instance %s, reason %s
Description	LDP has received notification messages. There are basically two types of notification, one is advisory and the other is fatal. In case a fatal notification is received, the session will be reset.
Log Module	ldp
Log Group	notification
Severity	INFO
Recommended Actions	If the peer is reset, run the show ldp session detail command and check the “Reason” attribute.

Log ID	LDP_NOTIFICATION_OUT
Message	LDP Notification sent, peer %s, instance %s, reason %s
Description	LDP has sent out notification messages to the peer.
Log Module	ldp
Log Group	notification
Severity	INFO
Recommended Actions	If peer is reset, run the show ldp session detail command and check the “Reason” attribute.

Log ID	LDP_SESSION_STATE
Message	LDP session state changed to %s, peer %s, instance %s
Description	LDP session state has changed
Log Module	ldp
Log Group	peer
Severity	INFO
Recommended Actions	None Required.

License Log Messages

Log ID	LICENSE_INTERNAL_NULL_STRING

Message	NULL license string passed to license verification API
Description	Invalid parameters passed for the license verification API.
Log Module	license
Log Group	internal
Severity	ERROR
Recommended Actions	Internal error. Contact RtBrick customer support

Log ID	LICENSE_INTERNAL_INVALID_STRING
Message	Invalid license string passed to license verification API
Description	Invalid parameters passed for the license verification API.
Log Module	license
Log Group	internal
Severity	INFO
Recommended Actions	Internal error. Contact RtBrick customer support

Log ID	LICENSE_FILE_READ_FAILURE
Message	Unable to read license string from license file %s
Description	License file read failure.
Log Module	license
Log Group	internal
Severity	ERROR
Recommended Actions	Internal error. Contact RtBrick customer support.

Log ID	LICENSE_FILE_OPEN_FAILURE
Message	Unable to open from license file %s
Description	Unable to open the license file.
Log Module	license
Log Group	internal

Severity	ERROR
Recommended Actions	Internal error. Contact RtBrick customer support

Log ID	LICENSE_STRING_ALLOCATION_FAILURE
Message	Unable to allocate %s bytes for license string
Description	Memory allocation failure
Log Module	license
Log Group	internal
Severity	ERROR
Recommended Actions	Contact RtBrick customer support

Log ID	LICENSE_INVALID_STRING_IN_FILE
Message	Invalid license string present in license file %s
Description	Invalid license installed
Log Module	license
Log Group	internal
Severity	ERROR
Recommended Actions	Contact RtBrick customer support

Log ID	LICENSE_INVALID_CONTENT
Message	Invalid license content in file %s
Description	Invalid license content
Log Module	license
Log Group	internal
Severity	ERROR
Recommended Actions	Contact RtBrick customer support

Log ID	LICENSE_DIRECTORY_OPEN_FAILURE

Message	Unable to open license directory
Description	License directory open failure.
Log Module	license
Log Group	internal
Severity	ERROR
Recommended Actions	Contact Rtbrick customer support

Log ID	LICENSE_START_DATE_MISSING
Message	License doesn’t have a start date
Description	License file did not have a start date.
Log Module	license
Log Group	internal
Severity	ERROR
Recommended Actions	Start date is missing in the license file. Obtain a valid license.

Log ID	LICENSE_END_DATE_MISSING
Message	License doesn’t have an end date
Description	License file did not have an end date.
Log Module	license
Log Group	internal
Severity	ERROR
Recommended Actions	End date is missing in the license file. Obtain a valid license.

Log ID	LICENSE_DIRECTORY_CLOSE_FAILURE
Message	Unable to close the license directory.
Description	Unable to close the license directory.
Log Module	license
Log Group	internal

Severity	ERROR
Recommended Actions	Contact RtBrick customer support

Log ID	LICENSE_HOSTCONFD_CONNECTION_DOWN
Message	License installation delayed due to hostconfd connection down
Description	License installation has been delayed due to the hostconfd connectivity issue.
Log Module	license
Log Group	internal
Severity	INFO
Recommended Actions	Contact RtBrick customer support

Log ID	LICENSE_INIT_FAILED
Message	License initialization failed.
Description	License initialization has failed.
Log Module	license
Log Group	internal
Severity	ERROR
Recommended Actions	Contact RtBrick customer support

Log ID	LICENSE_TABLE_NOT_PRESENT
Message	License table (global.rtbrick.license) is not present
Description	The license table global.rtbrick.license is not present.
Log Module	license
Log Group	internal
Severity	ERROR
Recommended Actions	Contact RtBrick customer support

Log ID	LICENSE_SECONDARY_VALIDATION_COMPROMISED
Message	License table (global.rtbrick.license) is not present
Description	The license table global.rtbrick.license is not present.
Log Module	license
Log Group	internal
Severity	ERROR
Recommended Actions	Contact RtBrick customer support

Log ID	LICENSE_STATS_TABLE_NOT_PRESENT
Message	License stats table (local.rtbrick.license.stats) is not present
Description	The license stats table local.rtbrick.license.stats is not present.
Log Module	license
Log Group	internal
Severity	ERROR
Recommended Actions	Contact Rtbrick customer support

Log ID	LICENSE_CONFIG_TABLE_CREATE_FAILED
Message	License config table (local.rtbrick.license.config) is not present
Description	License stats table local.rtbrick.license.config is not present.
Log Module	license
Log Group	internal
Severity	ERROR
Recommended Actions	Contact RtBrick customer support

Log ID	LICENSE_TRAIL_PERIOD_EXPIRY
Message	The software trial period has ended, install a license.
Description	The software trial period has ended.

Log Module	license
Log Group	operational
Severity	CRITICAL
Recommended Actions	Install a new license.

Log ID	LICENSE_EXPIRY_WARNING
Message	The current license will expire by %s, install a new license
Description	The current license is about to expire.
Log Module	license
Log Group	operational
Severity	WARNING
Recommended Actions	Install a new license.

Log ID	LICENSE_EXPIRY_ERROR
Message	The current license will expire by %s, please install a new license
Description	The current license is about to expire.
Log Module	license
Log Group	operational
Severity	ERROR
Recommended Actions	Install a new license.

Log ID	LICENSE_EXPIRY_CRITICAL
Message	The current license has expired, install a new license
Description	The current license has expired.
Log Module	license
Log Group	operational
Severity	CRITICAL

Recommended Actions	Install a new license.

Policy Log Messages

Log ID	POLICY_LOG_CODE_GEN_STATUS_FAILED
Message	Policy [%s]: Code Generation Error Reason [%s]
Description	Policy code generation has failed.
Log Module	policy
Log Group	Generation
Severity	ERROR
Recommended Actions	Delete the old policy library and if the problem still persists, contact RtBrick customer support.

Log ID	POLICY_LOG_CODE_GEN_STATUS_SUCCESS
Message	Policy [%s]: Code Generation Status: [%s]
Description	Policy code generation status
Log Module	policy
Log Group	Generation
Severity	INFO
Recommended Actions	No action required.

PPPoE Log Messages

Log ID	PPPOE_CONFIG_NOT_FOUND
Message	No matching configuration found for PPPoE discovery on interface %s, MAC %s, S-Tag %s, C-Tag %s
Description	Received PPPoE discovery packet without a matching configuration.
Log Module	pppoe
Log Group	config
Severity	info

Recommended Actions	Verify the access interface configuration.

Log ID	PPPOE_SUBSCRIBER_ID_GET_FAILED
Message	PPPoE session subscriber-id get failed on interface %s, MAC %s, PPPoE session-id %s, S-Tag %s, C-Tag %s
Description	Failure in assigning the subscriber identifier to the PPPoE session.
Log Module	pppoe
Log Group	subscriber
Severity	info
Recommended Actions	No action required.

Log ID	PPPOE_FSM_STATE_CHANGE
Message	%s session for subscriber-id %s changed state from %s to %s for input %s, action %s
Description	PPP FSM state has changed.
Log Module	pppoe
Log Group	ppp
Severity	info
Recommended Actions	No action required.

Log ID	PPPOE_PPP_SEND_ERROR
Message	Failed to encode and send PPP packet on interface %s, subscriber-id %s
Description	Failed to encode and send PPP packet.
Log Module	pppoe
Log Group	ppp
Severity	info

Recommended Actions	No action required.

Log ID	PPPOE_LCP_WRONG_ECHO_MAGIC
Message	LCP ECHO received on interface %s, subscriber-id %s with wrong magic number %s, expected %s
Description	PPP LCP echo request/reply received with a wrong magic number.
Log Module	pppoe
Log Group	ppp
Severity	info
Recommended Actions	Restart the subscriber.

Log ID	PPPOE_LCP_WRONG_ECHO_REPLY_ID
Message	LCP ECHO reply received on interface %s, subscriber-id %s with wrong identifier %s, expected %s
Description	PPP LCP echo reply received with a wrong identifier.
Log Module	pppoe
Log Group	ppp
Severity	info
Recommended Actions	Restart the subscriber.

Log ID	PPPOE_LCP_RCV_PROTO_REJECT
Message	PPP protocol reject received on interface %s, subscriber-id %s protocol %s
Description	PPP protocol 'reject' has been received.
Log Module	pppoe
Log Group	ppp
Severity	info

Recommended Actions	No action required.

Log ID	PPPOE_PROTO_DECODE_ERROR
Message	%s packet decode error on interface %s, MAC %s, session-id %s, subscriber-id %s with remaining packet length %s
Description	Failed to decode the received PPPoE PPP packet.
Log Module	pppoe
Log Group	ppp
Severity	info
Recommended Actions	No action required.

Log ID	PPPOE_LCP_RCV_WRONG_ID
Message	LCP %s received on interface %s, subscriber-id %s with identifier %s, expected %s
Description	Received PPP LCP response with a wrong identifier.
Log Module	pppoe
Log Group	ppp
Severity	info
Recommended Actions	No action required.

Log ID	PPPOE_LCP_RCV_MRU
Message	LCP %s received on interface %s, subscriber-id %s with received MRU %s, local MRU %s
Description	PPP LCP response with wrong or suggested MRU received.
Log Module	pppoe
Log Group	ppp
Severity	info
Recommended Actions	No action required.

Log ID	PPPOE_LCP_RCV_MAGIC
Message	LCP %s received on interface %s, subscriber-id %s with received magic %s, local magic %s
Description	PPP LCP response with wrong or suggested magic number received.
Log Module	pppoe
Log Group	ppp
Severity	info
Recommended Actions	Restart the subscriber.

Log ID	PPPOE_LCP_RCV_AUTH
Message	LCP %s received on interface %s, subscriber-id %s with wrong authentication protocol %s, expected %s
Description	PPP LCP response with wrong or suggested auth protocol received.
Log Module	pppoe
Log Group	ppp
Severity	info
Recommended Actions	Restart the subscriber.

Log ID	PPPOE_PPP_RCV_ERROR
Message	%s receive error %s on interface %s, MAC %s, session-id %s, subscriber-id %s
Description	Error in receiving PPP.
Log Module	pppoe
Log Group	ppp
Severity	info
Recommended Actions	Restart the subscriber.

Log ID	PPPOE_PROTO_MAX_RETRANSMISSION
Message	Maximum %s retransmission count on interface %s, subscriber-id %s, retransmit count %s
Description	PPPoE PPP retransmission has exceeded.
Log Module	pppoe
Log Group	ppp
Severity	info
Recommended Actions	Restart the subscriber.

Log ID	PPPOE_AUTH_REJECT
Message	Authentication rejected on interface %s, subscriber-id %s with user %s
Description	PPPoE PPP authentication has been rejected.
Log Module	pppoe
Log Group	ppp
Severity	info
Recommended Actions	Verify the subscriber username and password.

Log ID	PPPOE_NO_SOURCE_IP
Message	%s source IP address not found on interface %s, subscriber-id %s
Description	Source IP address is missing to set up PPPoE PPP session.
Log Module	pppoe
Log Group	ppp
Severity	info
Recommended Actions	Configure source IP for the source IFL.

Log ID	PPPOE_PROTO_RECEIVE_ERROR

Message	Protocol receive error %s on interface %s, MAC %s, session-id %s, subscriber-id %s
Description	Error in receiving PPPoE PPP protocol.
Log Module	pppoe
Log Group	ppp
Severity	info
Recommended Actions	No action required.

Log ID	PPPOE_PROTO_RCV_ID_WARNING
Message	%s %s received on interface %s, subscriber-id %s with wrong identifier %s, expected %s
Description	PPP NCP response received with a wrong identifier.
Log Module	pppoe
Log Group	ppp
Severity	info
Recommended Actions	No action required.

Log ID	PPPOE_IPCP_RECEIVE_WRONG_IP
Message	IPCP %s received on interface %s, subscriber-id %s with address %s, expected %s
Description	PPP IPCP reply received with wrong address.
Log Module	pppoe
Log Group	ppp
Severity	info
Recommended Actions	No action required.

Log ID	PPPOE_IPV6_CHECKSUM_ERROR
Message	PPPoE IPv6 packet decode checksum error on interface %s, subscriber-id %s

Description	Failed to decode the received PPPoE PPP packet with a wrong IPv6 checksum (UDP or ICMPv6) value.
Log Module	pppoe
Log Group	ppp
Severity	info
Recommended Actions	No action required.

Log ID	PPPOE_DECODE_ERROR
Message	Error decoding received PPPoE packet on interface %s, MAC %s
Description	Failed to decode the PPPoE ether type or VLAN.
Log Module	pppoe
Log Group	pppoe
Severity	info
Recommended Actions	No action required.

Log ID	PPPOE_DECODE_ERROR_WITH_VLAN
Message	Error decoding received PPPoE packet on interface %s, MAC %s, S-Tag %s, C-Tag %s
Description	Failed to decode the PPPoE discovery packet.
Log Module	pppoe
Log Group	pppoe
Severity	info
Recommended Actions	Verify the options and VLAN configured in the subscriber.

Log ID	PPPOE_UNKNOWN_DISCOVERY_PACKET
Message	Received unknown PPPoE discovery packet on interface %s, MAC %s, S-Tag %s, C-Tag %s, session-id %s
Description	Received unknown PPPoE discovery packet.

Log Module	pppoe
Log Group	pppoe
Severity	info
Recommended Actions	Check the options and VLAN configured in the subscriber.

Log ID	PPPOE_SESSION_CREATE
Message	PPPoE session created on interface %s, S-Tag %s, C-Tag %s, MAC %s, PPPoE session-id %s, subscriber-id %s, state %s access-profile %s
Description	Creation of the PPPoE/PPP session after PADS is sent.
Log Module	pppoe
Log Group	pppoe
Severity	info
Recommended Actions	No action required.

Log ID	PPPOE_SESSION_LIMIT_REACHED
Message	PPPoE session limit reached on interface %s, MAC %s, S-Tag %s, C-Tag %s
Description	PPPoE session limit reached.
Log Module	pppoe
Log Group	pppoe
Severity	notice
Recommended Actions	New sessions can be created only after removing existing sessions. Clear existing subscribers.

Log ID	PPPOE_PACKET_ENCODE_ERROR
Message	Could not encode PPPoE %s on interface %s, MAC %s, PPPoE session-id %s, S-Tag %s, C-Tag %s
Description	PPPoE discovery encode error.
Log Module	pppoe

Log Group	pppoe
Severity	info
Recommended Actions	No action required.

Log ID	PPPOE_BAD_PACKET_INTERFACE_GET_FAILED
Message	Interface get failed for PPPoE %s packet on interface %s, MAC %s
Description	PPPoE discovery packet has been received on the invalid interface.
Log Module	pppoe
Log Group	pppoe
Severity	info
Recommended Actions	Check the access interface configuration.

Log ID	PPPOE_BAD_PACKET_INTERFACE_DOWN
Message	Received PPPoE %s packet on operationaly down interface %s, MAC %s
Description	PPPoE discovery packet has been received on the interface that is down.
Log Module	pppoe
Log Group	pppoe
Severity	info
Recommended Actions	Verify the access interface state.

Log ID	PPPOE_BAD_PACKET_WRONG_MAC
Message	Received bad PPPoE %s packet with wrong MAC on interface %s, MAC %s
Description	Invalid PPPoE discovery packet has been received with the wrong MAC address.
Log Module	pppoe

Log Group	pppoe
Severity	info
Recommended Actions	Restart the subscriber.

Log ID	PPPOE_BAD_PACKET_NO_AC_COOKIE
Message	Received bad PPPoE %s packet on interface %s, MAC %s with no ac cookie
Description	Invalid PPPoE discovery packet has been received with no ac_cookie.
Log Module	pppoe
Log Group	pppoe
Severity	info
Recommended Actions	No action required.

Log ID	PPPOE_BAD_PACKET_WRONG_AC_COOKIE
Message	Received bad PPPoE %s packet on interface %s, MAC %s with wrong ac cookie
Description	Invalid PPPoE discovery packet has been received with the wrong ac_cookie.
Log Module	pppoe
Log Group	pppoe
Severity	info
Recommended Actions	No action required.

Log ID	PPPOE_BAD_PACKET_UNKNOWN_ETHER_TYPE
Message	Received bad PPPoE packet on interface %s, MAC %s with unknown ether type %s
Description	Received PPPoE discovery packet with an invalid ether type.
Log Module	pppoe

Log Group	pppoe
Severity	info
Recommended Actions	Verify the subscriber configuration.

Log ID	PPPOE_SESSION_DELETE
Message	PPPoE session delete on interface %s, MAC %s, PPPoE session-id %s, S-Tag %s, C-Tag %s
Description	PPPoE session deletion occurred.
Log Module	pppoe
Log Group	pppoe
Severity	info
Recommended Actions	No action required.

Log ID	PPPOE_DHCPV6_DECODE_ERROR
Message	PPPoE DHCPv6 packet decode error on interface %s, subscriber-id %s error code: %s subcode: %s
Description	Failed to decode the received PPPoE DHCPv6 packet.
Log Module	pppoe
Log Group	dhcpv6
Severity	info
Recommended Actions	No action required.

Log ID	PPPOE_DHCPV6_VALIDATION_ERROR
Message	PPPoE DHCPv6 packet validation error on interface %s, subscriber-id %s
Description	Failed to validate the received PPPoE DHCPv6 packet.
Log Module	pppoe
Log Group	dhcpv6
Severity	info

Recommended Actions	No action required.

Log ID	PPPOE_VENDOR_REQUEST_FAILED
Message	LCP vendor request failed for subscriber-id %s, %s
Description	PPPoE PPP LCP vendor request failure.
Log Module	pppoe
Log Group	ppp
Severity	info
Recommended Actions	No action required.

Log ID	PPPOE_RECEIVE_ERROR
Message	PPPOE packet receive with %s
Description	Generic PPPoE receive error.
Log Module	pppoe
Log Group	general
Severity	debug
Recommended Actions	No action required.

Log ID	PPPOE_CONFIG_FOUND
Message	PPPoE %s received on interface %s, MAC %s, S-Tag %s, C-Tag %s, found access-profile %s
Description	Received PPPoE discovery packet with a matching configuration.
Log Module	pppoe
Log Group	config
Severity	debug
Recommended Actions	No action required.

Log ID	PPPOE_SUBSCRIBER_IN
Message	Result from subscriberd for subscriber-id %s with action %s
Description	Action received from subscriberd.
Log Module	pppoe
Log Group	subscriber
Severity	debug
Recommended Actions	No action required.

Log ID	PPPOE_LCP_SESSION_EVENT
Message	LCP session %s on interface %s, MAC %s, PPPoE session-id %s, subscriber-id %s
Description	PPP LCP protocol events.
Log Module	pppoe
Log Group	ppp
Severity	debug
Recommended Actions	No action required.

Log ID	PPPOE_IPCP_SESSION_EVENT
Message	IPCP session %s on interface %s, MAC %s, PPPoE session-id %s, subscriber-id %s
Description	PPP IPCP protocol events.
Log Module	pppoe
Log Group	ppp
Severity	debug
Recommended Actions	No action required.

Log ID	PPPOE_IP6CP_SESSION_EVENT
Message	IP6CP session %s on interface %s, MAC %s, PPPoE session-id %s, subscriber-id %s

Description	PPP IP6CP protocol events
Log Module	pppoe
Log Group	ppp
Severity	debug
Recommended Actions	Contact Rtbrick customer support

Log ID	PPPOE_LCP_PRINT_MY_OPTIONS
Message	PPP LCP options for subscriber-id %s, MRU %s, magic %s, authentication %s, identifier %s, retransmission-count %s, retransmisson-interval %s
Description	Local PPP LCP options.
Log Module	pppoe
Log Group	ppp
Severity	debug
Recommended Actions	No action required.

Log ID	PPPOE_LCP_START_ECHO
Message	LCP start echo on interface %s, subscriber-id %s
Description	Started sending the PPP LCP echo request interval.
Log Module	pppoe
Log Group	ppp
Severity	debug
Recommended Actions	No action required.

Log ID	PPPOE_LCP_RCV_CONFX
Message	PPP LCP %s received on interface %s, subscriber-id %s, MRU %s, magic %s, authentication %s
Description	PPP LCP request/response has been received.
Log Module	pppoe

Log Group	ppp
Severity	debug
Recommended Actions	No action required.

Log ID	PPPOE_LCP_NEGOTIATION
Message	LCP %s on interface %s for subscriber-id %s
Description	PPP LCP negotiation occurred.
Log Module	pppoe
Log Group	ppp
Severity	debug
Recommended Actions	No action required.

Log ID	PPPOE_LCP_SETUP
Message	LCP setup %s on interface %s for subscriber-id %s
Description	PPPoE PPP sessions setup.
Log Module	pppoe
Log Group	ppp
Severity	debug
Recommended Actions	No action required.

Log ID	PPPOE_AUTH_RCV_REQ
Message	%s receive request packet on interface %s, subscriber-id %s, user %s
Description	PPPoE PPP authentication request has been received.
Log Module	pppoe
Log Group	ppp
Severity	debug
Recommended Actions	No action required.

Log ID	PPPOE_LCP_UPPPER_LAYER_TEARDOWN
Message	LCP teardown by upper layer on interface %s for subscriber-id %s
Description	PPPoE PPP session teardown/termination.
Log Module	pppoe
Log Group	ppp
Severity	debug
Recommended Actions	No action required.

Log ID	PPPOE_NCP_SESSION_DETAIL
Message	%s session on interface %s, subscriber-id %s, retransmission-interval %s, local ipv4 address %s
Description	PPPoE PPP NCP local session information.
Log Module	pppoe
Log Group	ppp
Severity	debug
Recommended Actions	No action required.

Log ID	PPPOE_IPCP_CONFREQ_PEER
Message	IPCP CONFREQ peer info for subscriber-id %s on interface %s, peer ipv4-address %s, primary dns %s, secondary dns %s
Description	PPPoE PPP IPCP session information.
Log Module	pppoe
Log Group	ppp
Severity	debug
Recommended Actions	No action required.

Log ID	PPPOE_IPCP_CONFREQ

Message	IPCP CONFREQ received for subscriber-id %s on interface %s, peer ipv4-address %s, primary dns %s, secondary dns %s
Description	PPPoE PPP IPCP has been received.
Log Module	pppoe
Log Group	ppp
Severity	debug
Recommended Actions	No action required.

Log ID	PPPOE_DISCOVERY_PACKET_RECEIVE
Message	Received PPPoE %s packet on interface %s, MAC %s, session-id %s
Description	Received PPPoE discovery packet.
Log Module	pppoe
Log Group	pppoe
Severity	debug
Recommended Actions	No action required.

Log ID	PPPOE_DISCOVERY_PACKET_SEND
Message	Sending PPPoE %s packet on interface %s, MAC %s, session-id %s
Description	Sent PPPoE discovery packet.
Log Module	pppoe
Log Group	pppoe
Severity	debug
Recommended Actions	No action required.

Log ID	PPPOE_DHCPV6_LEASE
Message	PPPoE DHCPv6 IA_PD %s for subscriber-id %s, MAC %s, S-Tag %s, C-Tag %s

Description	PPPoE DHCPv6 IA_PD lease.
Log Module	pppoe
Log Group	dhcpv6
Severity	debug
Recommended Actions	No action required.

Log ID	PPPOE_FSM_ACTION_FAILED
Message	%s session for subscriber-id %s, MAC %s, S-Tag %s, C-Tag %s action %s failed
Description	PPP FSM action failure.
Log Module	pppoe
Log Group	ppp
Severity	debug
Recommended Actions	No action required.

Log ID	PPPOE_PADT_RECEIVED
Message	PPPoE PADT received for subscriber-id %s on interface %s, MAC %s, S-Tag %s, C-Tag %s
Description	Received PPPoE padt packet.
Log Module	pppoe
Log Group	config
Severity	debug
Recommended Actions	No action required.

Subscriber Management Log Messages

Log ID	SUBSMGMT_AAA_PROFILE_NOT_FOUND
Message	AAA configuration profile %s not found for subscriber %s

Description	This event log is generated if the authentication, authorization, and accounting (AAA) configuration profile for the subscriber has not been found.
Log Module	subsMgmt
Log Group	config
Severity	warning
Recommended Actions	Verify the AAA profile configuration.

Log ID	SUBSMGMT_AAA_PROFILE_INVALID_ORDER
Message	AAA configuration profile %s invalid configuration order for subscriber %s
Description	AAA configuration profile is not in valid order.
Log Module	subsMgmt
Log Group	config
Severity	info
Recommended Actions	Verify the authentication and accounting order in the AAA profile configuration.

Log ID	SUBSMGMT_RADIUS_PROFILE_MISSING_IN_AAA
Message	RADIUS profile missing in AAA profile %s for subscriber %s
Description	Missing RADIUS profile in AAA profile.
Log Module	subsMgmt
Log Group	config
Severity	warning
Recommended Actions	Verify the RADIUS profile in the AAA profile configuration.

Log ID	SUBSMGMT_MISSING_AAA_CONFIG
Message	missing AAA %s for subscriber %s
Description	Subscriber AAA configuration warning.
Log Module	subsMgmt

Log Group	config
Severity	warning
Recommended Actions	Configure missing AAA configuration.

Log ID	SUBSMGMT_AUTHENTICATION_FAILED
Message	Authentication failed for subscriber %s due to %s
Description	Subscriber authentication failure.
Log Module	subsMgmt
Log Group	subscriber
Severity	info
Recommended Actions	Check the local/radius authentication configuration.

Log ID	SUBSMGMT_ACCT_SESSION_ID_NOT_FOUND
Message	No subscriber with Acct-Session-ID %s found
Description	a CoA request with an unknown Acct-Session-ID has been received.
Log Module	subsMgmt
Log Group	subscriber
Severity	info
Recommended Actions	No action required.

Log ID	SUBSMGMT_SUBSCRIBER_ADD
Message	Subscriber %s added
Description	New subscriber has been added.
Log Module	subsMgmt
Log Group	subscriber
Severity	info
Recommended Actions	No action required.

Log ID	SUBSMGMT_SUBSCRIBER_DEL
Message	Subscriber %s deleted
Description	Subscriber has been deleted.
Log Module	subsMgmt
Log Group	subscriber
Severity	info
Recommended Actions	No action required.

Log ID	SUBSMGMT_SUBSCRIBER_NOT_FOUND
Message	Subscriber %s not found
Description	Subscriber is not found.
Log Module	subsMgmt
Log Group	subscriber
Severity	info
Recommended Actions	No action required.

Log ID	SUBSMGMT_TRAP_ADD
Message	Failed to add I/O trap %s
Description	Failed to add input/output trap entry.
Log Module	subsMgmt
Log Group	subscriber
Severity	notice
Recommended Actions	Check the FIB and RIB logs.

Log ID	SUBSMGMT_TRAP_DEL
Message	Failed to delete I/O trap %s
Description	Failed to delete input/output trap entry.
Log Module	subsMgmt

Log Group	subscriber
Severity	info
Recommended Actions	Check the FIB and RIB logs.

Log ID	SUBSMGMT_QOS_SHAPER
Message	Invalid dynamic shaper definition %s for subscriber %s
Description	Invalid subscriber QoS shaper definition has been received.
Log Module	subsMgmt
Log Group	subscriber
Severity	notice
Recommended Actions	No action required.

Log ID	SUBSMGMT_QOS_POLICER
Message	Invalid dynamic policer definition %s for subscriber %s
Description	Invalid subscriber QoS policer definition has been received.
Log Module	subsMgmt
Log Group	subscriber
Severity	notice
Recommended Actions	No action required.

Log ID	SUBSMGMT_IPV4_DUPLICATE_ADDRESS_DETECTED
Message	Duplicate IPv4 address %s detected for subscriber %s
Description	Duplicate IPv4 address has been detected.
Log Module	subsMgmt
Log Group	subscriber
Severity	notice
Recommended Actions	Verify the IPv4 address allocation configuration.

Log ID	SUBSMGMT_IPV6_DUPLICATE_ADDRESS_DETECTED
Message	Duplicate IPv6 prefix %s detected for subscriber %s
Description	Duplicate IPv6 address has been detected.
Log Module	subsMgmt
Log Group	subscriber
Severity	notice
Recommended Actions	Verify the IPv6 address allocation configuration.

Log ID	SUBSMGMT_INSTANCE_NOT_FOUND
Message	Instance %s not found for subscriber %s
Description	Instance is not found.
Log Module	subsMgmt
Log Group	subscriber
Severity	notice
Recommended Actions	Configure the subscriber instance.

Log ID	SUBSMGMT_FSM_EVENT
Message	Subscriber %s FSM State change, from %s, event %s, to %s
Description	Subscriber FSM state transition.
Log Module	subsMgmt
Log Group	subscriberfsm
Severity	info
Recommended Actions	No action required.

Log ID	SUBSMGMT_POOL_IPV4_EXHAUSTED
Message	IPv4 pool %s exhausted for subscriber %s
Description	Local IPv4 address pool has exhausted.
Log Module	subsMgmt

Log Group	pool
Severity	notice
Recommended Actions	Free up IPv4 addresses.

Log ID	SUBSMGMT_POOL_IPV6_EXHAUSTED
Message	IPv6 pool %s exhausted for subscriber %s
Description	Local IPv6 address pool has exhausted.
Log Module	subsMgmt
Log Group	pool
Severity	notice
Recommended Actions	Free up IPv6 addresses.

Log ID	SUBSMGMT_QOS_QUEUE
Message	Invalid dynamic queue definition %s for subscriber %s
Description	Invalid subscriber QoS queue definition has been received.
Log Module	subsMgmt
Log Group	subscriber
Severity	notice
Recommended Actions	No action required.

Log ID	SUBSMGMT_FUNCTION_INVOCATION
Message	Subscriber Daemon %s Function invoked
Description	Function calls in SubscriberD.
Log Module	subsMgmt
Log Group	general
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_SERVICE_PROFILE_GET_FAILED
Message	Failed to get service profile %s for subscriber %s
Description	Failed to get service configuration profile.
Log Module	subsMgmt
Log Group	config
Severity	debug
Recommended Actions	Reconfigure the service profile.

Log ID	SUBSMGMT_INTERNAL_AUTHENTICATION
Message	Failed to %s during authentication for subscriber %s
Description	General subscriber authentication warning.
Log Module	subsMgmt
Log Group	subscriber
Severity	warning
Recommended Actions	No action required.

Log ID	SUBSMGMT_INTERNAL_ACCOUNTING
Message	Failed to %s during accounting for subscriber %s
Description	General subscriber accounting warning.
Log Module	subsMgmt
Log Group	subscriber
Severity	warning
Recommended Actions	No action required.

Log ID	SUBSMGMT_PPPOE_RESULT
Message	Failed to %s from PPPoE for subscriber %s
Description	General subscriber PPPoE related errors.
Log Module	subsMgmt

Log Group	subscriber
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_L2TP_RESULT
Message	Failed to extract the L2TP result
Description	General subscriber L2TP related errors.
Log Module	subsMgmt
Log Group	subscriber
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_ACCOUNTING_STOP_PROGRESS
Message	Accounting STOP in progress for subscriber %s
Description	Subscriber waits for the final counter update.
Log Module	subsMgmt
Log Group	subscriber
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_IFL_CREATE
Message	Failed to create %s for subscriber %s
Description	Subscriber logical interface errors.
Log Module	subsMgmt
Log Group	subscriber
Severity	info
Recommended Actions	No action required.

Log ID	SUBSMGMT_IFL_DELETE
Message	Failed to delete %s for ifl %s
Description	Subscriber logical interface delete event.
Log Module	subsMgmt
Log Group	subscriber
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_QOS_CREATE
Message	QoS settings created for subscriber %s with IFL %s
Description	Add subscriber QoS settings.
Log Module	subsMgmt
Log Group	subscriber
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_QOS_RESET
Message	QoS settings reset for subscriber %s with IFL %s
Description	Reset subscriber QoS settings.
Log Module	subsMgmt
Log Group	subscriber
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_QOS_UPDATE
Message	QoS settings updated for subscriber %s with IFL %s
Description	Subscriber QoS setting is updated.
Log Module	subsMgmt

Log Group	subscriber
Severity	debug
Recommended Actions	No Action required.

Log ID	SUBSMGMT_BDS_TABLE_CREATED
Message	BDS table %s created
Description	BDS table has been created.
Log Module	subsMgmt
Log Group	bds
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_BDS_TABLE_CREATE_ERROR
Message	Failed to create BDS table %s
Description	Failed to create BDS table.
Log Module	subsMgmt
Log Group	bds
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_BDS_TABLE_GET_ERROR
Message	Failed to get BDS table %s
Description	Failed to receive BDS table or the BDS table is not found.
Log Module	subsMgmt
Log Group	bds
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_BDS_TEMPLATE_GET_ERROR
Message	Failed to get BDS object template %s
Description	Failed to receive the BDS object template.
Log Module	subsMgmt
Log Group	bds
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_BDS_TEMPLATE_POPULATE_ERROR
Message	Failed to populate BDS object template %s
Description	Failed to populate BDS object template.
Log Module	subsMgmt
Log Group	bds
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_BDS_OBJECT_ADD_ERROR
Message	Failed to add BDS object to table %s
Description	Failed to add BDS object to the table.
Log Module	subsMgmt
Log Group	bds
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_BDS_OBJECT_DEL_ERROR
Message	Failed to delete BDS object from table %s
Description	Failed to delete BDS object from the table.
Log Module	subsMgmt

Log Group	bds
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_RADIUS_PROFILE_ACTIVATE
Message	Failed to activate RADIUS profile
Description	RADIUS profile activation event.
Log Module	subsMgmt
Log Group	radius
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_RADIUS_ACCOUNTING_QUEUE_INDEX
Message	RADIUS accounting queue index exhausted
Description	RADIUS accounting queue index
Log Module	subsMgmt
Log Group	radius
Severity	debug
Recommended Actions	Contact Rtbrick customer support

Log ID	SUBSMGMT_RADIUS_AUTHENTICATION_QUEUE_INDEX
Message	RADIUS authentication queue index is exhausted
Description	RADIUS accounting queue index exhausted.
Log Module	subsMgmt
Log Group	radius
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_RADIUS_ACCOUNTING_EVENT
Message	Failed to %s RADIUS accounting request from %s
Description	RADIUS accounting event.
Log Module	subsMgmt
Log Group	radius
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_RADIUS_PROFILE_ADDED
Message	RADIUS Profile %s added
Description	Added a RADIUS profile.
Log Module	subsMgmt
Log Group	radius
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_RADIUS_PROFILE_DELETED
Message	RADIUS profile %s deleted
Description	RADIUS profile has been deleted.
Log Module	subsMgmt
Log Group	radius
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_RADIUS_PROFILE_NOT_FOUND
Message	RADIUS profile %s not found
Description	RADIUS profile has not been found.
Log Module	subsMgmt

Log Group	radius
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_RADIUS_SERVER_ADDED
Message	RADIUS server %s added
Description	RADIUS server has been added.
Log Module	subsMgmt
Log Group	radius
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_RADIUS_SERVER_DELETED
Message	RADIUS server %s deleted
Description	RADIUS server has been deleted.
Log Module	subsMgmt
Log Group	radius
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_RADIUS_SERVER_DELETE_FAILED
Message	Failed to delete RADIUS server %s
Description	RADIUS server deletion failure.
Log Module	subsMgmt
Log Group	radius
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_RADIUS_SERVER_NOT_FOUND
Message	RADIUS server %s not found
Description	RADIUS server is not found.
Log Module	subsMgmt
Log Group	radius
Severity	debug
Recommended Actions	Configure RADIUS server.

Log ID	SUBSMGMT_RADIUS_SERVER_ACCOUNTING_DISABLED
Message	RADIUS server %s accounting disabled
Description	RADIUS server accounting has been disabled.
Log Module	subsMgmt
Log Group	radius
Severity	debug
Recommended Actions	Check the RADIUS server configuration.

Log ID	SUBSMGMT_RADIUS_SERVER_AUTHENTICATION_DISABLED
Message	RADIUS server %s authentication disabled
Description	RADIUS server authentication has been disabled.
Log Module	subsMgmt
Log Group	radius
Severity	debug
Recommended Actions	Check the RADIUS server configuration.

Log ID	SUBSMGMT_RADIUS_SERVER_UNKNOWN
Message	RADIUS message from unknown server %s
Description	RADIUS requests from unknown IPv4 addresses.
Log Module	subsMgmt

Log Group	radius
Severity	debug
Recommended Actions	Check the RADIUS server configuration.

Log ID	SUBSMGMT_RADIUS_SERVER_GET_FAILED
Message	RADIUS server %s get failed while decoding RADIUS message %s
Description	RADIUS server failure.
Log Module	subsMgmt
Log Group	radius
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_RADIUS_TUNNEL_ERROR
Message	Tunnel error %s for subscriber %s
Description	RADIUS subscriber tunnel-related errors.
Log Module	subsMgmt
Log Group	radius
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_RADIUS_AUTHENTICATION_ERROR
Message	Authentication error %s for subscriber %s
Description	RADIUS subscriber tunnel-related errors.
Log Module	subsMgmt
Log Group	radius
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_RADIUS_ACCOUNTING_ERROR
Message	Accounting error %s for subscriber %s
Description	RADIUS subscriber accounting error.
Log Module	subsMgmt
Log Group	radius
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_RADIUS_ENCODE_ERROR
Message	Encode error %s for subscriber %s
Description	RADIUS encode error.
Log Module	subsMgmt
Log Group	radius
Severity	debug
Recommended Actions	Check RADIUS configurations.

Log ID	SUBSMGMT_RADIUS_AUTH_REQ
Message	Added RADIUS authentication request for subscriber %s from server %s
Description	RADIUS authentication request.
Log Module	subsMgmt
Log Group	radius
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_RADIUS_AUTH_REQ_ERROR
Message	Failed to add RADIUS authentication request for subscriber %s from server %s
Description	RADIUS authentication request failure.

Log Module	subsMgmt
Log Group	radius
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_RADIUS_ACCOUNTING_REQUEST
Message	Added RADIUS accounting request for subscriber %s from server %s
Description	RADIUS accounting request.
Log Module	subsMgmt
Log Group	radius
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_RADIUS_ACCOUNTING_REGUEST_ERROR
Message	Failed to add RADIUS accounting request for subscriber %s from server %s
Description	RADIUS accounting request failure.
Log Module	subsMgmt
Log Group	radius
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_RADIUS_AUTH_STATE_CHANGE
Message	RADIUS Server %s Authentication State changed from %s to %s
Description	State change in RADIUS server authentication.
Log Module	subsMgmt
Log Group	radius

Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_RADIUS_ACCT_STATE_CHANGE
Message	RADIUS Server %s Accounting State changed from %s to %s
Description	State change in RADIUS server accounting.
Log Module	subsMgmt
Log Group	radius
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_RADIUS_PROFILE_STATE_CHANGE
Message	RADIUS Profile %s State changed from %s to %s
Description	State change in RADIUS profile accounting.
Log Module	subsMgmt
Log Group	radius
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_RADIUS_PROFILE_ERROR
Message	RADIUS profile %s %s for subscriber %s
Description	RADIUS profile subscriber event.
Log Module	subsMgmt
Log Group	radius
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_RADIUS_DROPPED

Message	RADIUS message from %s dropped due to %s
Description	RADIUS message has been dropped.
Log Module	subsMgmt
Log Group	radius
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_RADIUS_ASCEND_DATA_FILTER
Message	Ascend Data Filter error for subscriber %s due to %s
Description	RADIUS filter message decode error.
Log Module	subsMgmt
Log Group	radius
Severity	notice
Recommended Actions	No action required.

Log ID	SUBSMGMT_RADIUS_ASCEND_DATA_FILTER_NOT_SUPPORTED
Message	Ascend Data Filter attribute %s with value %x not supported for subscriber %s
Description	RADIUS filter message decode error.
Log Module	subsMgmt
Log Group	radius
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_FSM_ERROR
Message	FSM Error for subscriber %s, State %s, Event %s
Description	Invalid event for the current state of subscriber FSM.
Log Module	subsMgmt

Log Group	subscriberfsm
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_FSM_NOT_FOUND
Message	Subscriber %s not found for FSM event %s
Description	Subscriber has not been found for the received FSM event.
Log Module	subsMgmt
Log Group	subscriberfsm
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_FSM_STATIC_IP_ALLOCATION
Message	Static IPv4 address from access-profile allocated for subscriber %s
Description	Subscriber FSM event logging.
Log Module	subsMgmt
Log Group	subscriberfsm
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_FSM_GENERAL_ERROR
Message	%s for subscriber %s
Description	Subscriber FSM Error logging.
Log Module	subsMgmt
Log Group	subscriberfsm
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_POOL_CONFIG
Message	Failed to process pool configuration
Description	Subscriber management Pool warning.
Log Module	subsMgmt
Log Group	pool
Severity	debug
Recommended Actions	Check the address pool configuration.

Log ID	SUBSMGMT_POOL_ASSIGN_IPV4
Message	Failed to assign ipv4 address for subscriber %s
Description	Subscriber management pool warning.
Log Module	subsMgmt
Log Group	pool
Severity	debug
Recommended Actions	Increase address pool size.

Log ID	SUBSMGMT_POOL_RELEASE_IPV4
Message	Failed to release ipv4 address for subscriber %s
Description	Subscriber management pool warning.
Log Module	subsMgmt
Log Group	pool
Severity	debug
Recommended Actions	Increase address pool size.

Log ID	SUBSMGMT_POOL_ASSIGN_IPV6
Message	Failed to assign ipv6 address for subscriber %s
Description	Subscriber management pool warning.
Log Module	subsMgmt

Log Group	pool
Severity	debug
Recommended Actions	Increase address pool size.

Log ID	SUBSMGMT_POOL_RELEASE_IPV6
Message	Failed to release ipv6 address for subscriber %s
Description	Subscriber management pool warning.
Log Module	subsMgmt
Log Group	pool
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_POOL_IPV4_SUCCESS
Message	IPv4 address %s from pool %s assigned to subscriber %s
Description	IPv4 address from the local address pool has been assigned.
Log Module	subsMgmt
Log Group	pool
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_POOL_IPV6_SUCCESS
Message	IPv6 prefix %s from pool %s assigned to subscriber %s
Description	IPv6 prefix from local address pool assigned.
Log Module	subsMgmt
Log Group	pool
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_TEST_AAA_RESULT
Message	Failed to %s from test aaa request for subscriber %s
Description	General test AAA requests related errors.
Log Module	subsMgmt
Log Group	subscriber
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_IPOE_RESULT
Message	Failed to %s from IPoE for subscriber %s
Description	General subscriber IPoE related errors.
Log Module	subsMgmt
Log Group	subscriber
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_L2BSA_GENERIC_ERROR
Message	%s
Description	General subscriber L2BSA related errors.
Log Module	subsMgmt
Log Group	subscriber
Severity	error
Recommended Actions	No action required.

Log ID	SUBSMGMT_L2BSA_RESULT
Message	Failed to %s from L2BSA service for subscriber %s
Description	General L2BSA subscriber related errors.
Log Module	subsMgmt

Log Group	l2bsa
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_PLATFORM_TYPE
Message	Platform type set to %s
Description	Platform type has been updated.
Log Module	subsMgmt
Log Group	general
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_RD_INVALID_OBJECT
Message	Redundancy object with invalid subscriber-id/rd-session-id
Description	Redundancy invalid object.
Log Module	subsMgmt
Log Group	redundancy
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_RD_INVALID_EVENT
Message	Invalid event on redundancy session %s, %s
Description	Redundancy invalid event.
Log Module	subsMgmt
Log Group	redundancy
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_RD_SUBSCRIBER_EVENT
Message	Event for subscriber %s on redundancy session %s %s
Description	Redundancy subscriber event.
Log Module	subsMgmt
Log Group	redundancy
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_RD_ADD_FAILED
Message	Failed to add subscriber %s for redundancy session %s
Description	Failure in redundancy sync.
Log Module	subsMgmt
Log Group	redundancy
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_RD_OBJECT_VALIDATION_FAILED
Message	Object validation failed for subscriber %s in redundancy session %s due to %s
Description	Redundancy invalid object.
Log Module	subsMgmt
Log Group	redundancy
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_FRAMED_ROUTE
Message	Framed route error for subscriber %s due to %s
Description	Error in RADIUS framed route handling.

Log Module	subsMgmt
Log Group	radius
Severity	notice
Recommended Actions	No action required.

Log ID	SUBSMGMT_LI_SUBSCRIBER_NOT_FOUND
Message	Subscriber %s not found, Lawful Intercept request is aborted
Description	The lawful intercept subscriber has not been found.
Log Module	subsMgmt
Log Group	lawful-intercept
Severity	info
Recommended Actions	No action required.

Log ID	SUBSMGMT_LI_IFL_NOT_FOUND
Message	Subscriber IFL not found for %s , Lawful Intercept request is aborted
Description	The lawful intercept subscriber IFL has not been found.
Log Module	subsMgmt
Log Group	lawful-intercept
Severity	info
Recommended Actions	No action required.

Log ID	SUBSMGMT_LI_INVALID_ATTRIBUTE
Message	Invalid Lawful Intercept attribute %s for subscriber %s
Description	Subscriber Lawful Intercept error.
Log Module	subsMgmt
Log Group	lawful-intercept
Severity	info

Recommended Actions	No action required.

Log ID	SUBSMGMT_LI_REQUEST_FAILURE
Message	Failed to process the Lawful Intercept request for subscriber %s
Description	Subscriber Lawful Intercept error.
Log Module	subsMgmt
Log Group	lawful-intercept
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_LI_DECRYPTION_FAILED
Message	RADIUS Lawful Intercept request attribute %s decryption failed %s for subscriber %s
Description	Subscriber Lawful Intercept error.
Log Module	subsMgmt
Log Group	lawful-intercept
Severity	info
Recommended Actions	No action required.

Log ID	SUBSMGMT_LI_MISSING_SUBSCRIBER_ID
Message	Mandatory subscriber_id is missing, Lawful Intercept request is aborted
Description	Subscriber Lawful Intercept error.
Log Module	subsMgmt
Log Group	lawful-intercept
Severity	info
Recommended Actions	No action required.

Log ID	SUBSMGMT_LI_BDS_OBJECT_ADD_ERROR
Message	Failed to add BDS object to table %s
Description	Failed to add BDS object to the table.
Log Module	subsMgmt
Log Group	lawful-intercept
Severity	debug
Recommended Actions	No action required.

Log ID	SUBSMGMT_LI_BDS_OBJECT_DEL_ERROR
Message	Failed to delete BDS object from table %s
Description	Failed to delete BDS object from table.
Log Module	subsMgmt
Log Group	lawful-intercept
Severity	debug
Recommended Actions	No action required.

15.3. LED Control

15.3.1. LED Overview

You can use the switch LEDs to monitor the activity and performance of a network switch or router. Using the LED control functionality, you can control the LEDs that are available in the hardware platforms supported by RBFS.

There are two types of LEDs available in the switch:

- Network Port LEDs
- System LEDs

Supported Platforms

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

15.3.2. LED Definitions

Port LED Definition for UfiSpace S9500-22XST

The network port LEDs are used to show the status of the link associated with the LED. In a UfiSpace S9500-22XST switch, there are 12*10G SFP+ ports, 8*25G SFP28 ports, and 2*100G QSFP28 ports on the front panel. The SFP, SFP28, and QSFP28 ports operate in a full duplex mode when the speed is 10Gbps, 25Gbps, or 100Gbps.

LED	Condition	Status
SFP+ Port LED	On (Green)	Link is up
	Off	No link
SFP28 Port LED	On (Green)	Link is up
	Off	No link
QSFP28 Port LED	On (Green)	Link is up
	Off	No link

System LED Definition for UfiSpace S9500-22XST

The system LEDs are used to indicate the status of power and system.

LED Indicators	Condition	Status
Power	Off	No power or in shut down mode.
	Solid Green	System power is good and the BMC heating is complete.
	Blinking Green	System power is good and the BMC heating is in progress.
	Solid Yellow	System power is good but BMC heating is failed.
	Blinking Yellow	System power failure.

LED Indicators	Condition	Status
STAT (System Status)	Off	System (X86 and BMC) is no booted.
	Solid Green	System boot is complete.
	Blinking Green	System boot is in progress.
	Solid Yellow	Reserved
	Blinking Yellow	Reserved
PSU LED Functions and State	Off	No DC power to any of the PSUs.
	Flashing Red	No DC power to this particular PSU.
	Flashing Green	DC is present, only standby output is on. Poor contact.
	Green	PSU DC output is on and is normal.
	Red	PSU failure.
	Flashing between Green and Red	Warning. Working condition is not satisfied. Check the voltage, electric current, and temperature.
FAN LED Functions and State	Off	Main board 3.3V power is failed or fan is not present.
	Solid Green	Fan is present and interrupt de-assert.
	Blinking Green	NA
	Solid Yellow	NA
	Blinking Yellow	Fan is present but interrupt assert.

Port LED Definition for UfiSpace S9600-32X

The network port LEDs are used to show the status of the link associated with the LED. In a UfiSpace S9600-32X switch, there are 4*10G SFP+ ports and 32*100G QSFP28 ports on the front panel. The SFP+ and QSFP28 ports operate in full duplex mode when the speed is 10Gbps and 100Gbps respectively.

LED	Condition	Status
SFP+ Port LED	On (Green)	Link is up
	Off	No link
QSFP28 Port LED	On (Green)	Link is up
	Off	No link

System LED Definition for UfiSpace S9600-32X

The system LEDs are used to indicate the status of power and system.

LED	Condition	Status
SYS	Off	System (x86 and BMC) is not booted.
	Solid Green	System boot is complete.
	Blinking Green	System boot is in progress.
	Fast Blinking Green	Reserved.
	Solid Amber	Power is up but system boot is failed.
	Blinking Amber	Reserved
	Fast Blinking Amber	Reserved

LED	Condition	Status
FAN	Off	Main board 3.3V_FAN power failure or no power.
	Solid Green	All fans are working normally and interrupt de-assert.
	Blinking Green	Fans PWMs do not match or each one fan no present
	Fast Blinking Green	Reserved.
	Solid Amber	Each one of fans is failure (PWM=0)
	Blinking Amber	Fan is present but interrupt assert.
	Fast Blinking Amber	Reserved
PSU1	Off	No DC power or standby is failed.
	Solid Green	PSU DC output is ON and standby is normal.
	Blinking Green	No PSU is present.
	Fast Blinking Green	Reserved.
	Solid Amber	PSU DC output is failed and standby is normal.
	Blinking Amber	Warning. Working condition is not satisfied. (Check the voltage, electric current, and temperature).
	Fast Blinking Amber	Reserved

LED	Condition	Status
PSU2	Off	No DC power or standby is failed.
	Solid Green	PSU DC output is ON and standby is normal.
	Blinking Green	No PSU is present.
	Fast Blinking Green	Reserved.
	Solid Amber	PSU DC output is failed and standby is normal.
	Blinking Amber	Warning. Working condition is not satisfied. (Check the voltage, electric current, and temperature).
	Fast Blinking Amber	Reserved
ID	Off	Turned off location of function
	Solid Blue	Ready for location of function
	Blinking Blue	Location of switch in telcom center
	Fast Blinking Blue	Reserved.

Port LED Definition for UfiSpace S9600-72XC

The network port LEDs are used to show the status of the link associated with the LED. In an UfiSpace S9600-72XC switch, there are 64*25G SFP28 ports and 8*100G QSFP28 on the front panel. SFP28 and QSFP28 operate in full duplex mode when the speed is 25Gbps and 100Gbps respectively.

LED	Condition	Status
SFP28 Port LED	On (Green)	Link is up
	Off	No link

LED	Condition	Status
QSFP28 Port LED	On (Green)	Link is up
	Off	No link

System LED Definition for UfiSpace S9600-72XC

The system LEDs are used to indicate the status of power and system.

LED	Condition	Status
SYS	Off	No power
	Solid Green	Host CPU/BMC boot is complete
	Solid Amber	Power is up, but host CPU/BMC boot is failed.
FAN	Off	Fans are not initialized.
	Solid Green	All fans are working normally.
	Blinking Amber	Fan is not working. One or more fans need service.
PSU1	Off	No power
	Solid Green	PSU1 is working normally.
	Blinking Amber	PSU1 is failed. Service required.
PSU2	Off	No power
	Solid Green	PSU2 is working normally.
	Blinking Amber	PSU2 is failed. Service is needed.

Port LED Definition for UfiSpace S9510-28DC

The network port LEDs are used to represent the status of the link associated with the LED.

Port type	Condition	Status
SFP+	Up	Green
	Down	Off

Port type	Condition	Status
SFP28	Up	Green
	Down	Off
QSFP28	Up	Green
	Down	Off

System LED Definition for UfiSpace S9510-28DC

The system LEDs are used to indicate the status of power and system.

LED Indicators	Behavior	Status
Power	Off	No power or in shut down mode.
	Solid Green	System power is good.
	Blinking Green	System power is good but BMC power is failed.
	Solid Yellow	System power is good but CPU power is failed.
	Blinking Yellow	System power failure.
STAT (System Status)	Off	System (X86 and BMC) is not booted.
	Solid Green	System boot is complete.
	Blinking Green	System boot is in progress.
	Solid Yellow	System boot is complete. (DIAG OS)
	Blinking Yellow	Reserved

LED Indicators	Behavior	Status
PSU LED Functions and State	Off	No DC power to any of the PSUs.
	Flashing Red	No DC power to this particular PSU.
	Flashing Green	DC is present, only standby output is on. Poor contact.
	Green	PSU output is on and normal.
	Red	PSU failure.
	Flashing between Green and Red	Warning. Working condition is not satisfied. Check the voltage, electric current, and temperature.
FAN LED Functions and State	Off	Main board 3.3V power is failed or fan is not present.
	Solid Green	Fan is present but interrupt assert
	Blinking Green	NA
	Solid Yellow	NA
	Blinking Yellow	Fan is present but interrupt assert.

Port LED Definition for EdgeCore AGR130

The network port LEDs are used to represent the status of the link associated with the LED. In an Accton/EdgeCore AGR130 switch, there are 48*10G SFP+ ports and 6*100G QSFP28 on the front panel, the all SFP/QSFP28 will operate in full duplex mode when the speed is 10Gbps or 100Gbps.

LED	Condition	Status
SFP+ Port LED	On/Flashing Green	Link at 10G, flashing indicates activity
	On/Flashing Amber	Link at 1G, flashing indicates activity
	Off	No link
QSFP28 Port LED in 100G Mode (Port 49 - 54)	On/Flashing Green	Link at 100G, flashing indicates activity
	Off	No link
QSFP28 Port LED in 25G Breakout Mode	On/Flashing Amber	Link at 25G in breakout mode, flashing indicates activity
	Off	No link
QSFP28 Port LED in 40G Mode (Port 49 - 54)	On/Flashing Blue	Link at 40G mode, flashing indicates activity
	Off	No link
QSFP28 Port LED in 10G Breakout Mode	On/Flashing Purple	Link at 10G in breakout mode, flashing indicates activity
	Off	No link

System LED Definition for Edgecore AGR130

The system LEDs are used to indicate the status of power and system.

LED	Condition	Status
PSU1 (Power Supply Status)	Green	Power is operating normally
	Amber	Power supply is present but not powered on or faulty.
	Off	Power supply is not present.

LED	Condition	Status
PSU2 (Power Supply Status)	Green	Power is operating normally
	Amber	Power supply is present but not powered on or faulty.
	Off	Power supply is not present.
Diag (Diagnostic)	Green	System self-diagnostic test is successfully completed.
	Amber	System self-diagnostic test detected a fault.
FAN	Green	System FAN is operating normally
	Amber	Fan tray is present buy system FAN is faulty.
	Off	System is off

Port LED Definition for EdgeCore AGR420

The network port LEDs are used to represent the status of the link associated with the LED.

Port type	Condition	Status
SFP+	Up	Blue LED is on (2 LEDs per port)
	Down	Off
SFP28	Up	Blue LED is on (2 LEDs per port)
	Down	Off
QSFP28	Up	Blue LED is on (2 LEDs per port)
	Down	Off

System LED Definition for Edgecore AGR420

The system LEDs are used to indicate the status of power and system.

LED	Condition	Status
DIAG (Diagnostic)	Green	System self diagnostic test is successfully completed.
	Amber	System self-diagnostic test has detected a fault. (Faulty Fan, thermal, or any interface.)
LOC	Flashing Amber	Flashing by remote management command. Assists technician in finding the right device for service in the rack.
	Off	Not the switch that technician needs to find.
FAN	Green	System fan is working normally.
	Amber	Fan tray is present but system fan is faulty.
	Off	System is off.
PSU0 (power supply status)	Green	The power supply, #0, is working normally.
	Amber	Power is present, but not ON. Power is faulty.
	Off	Power supply is not present.
PSU1 (power supply status)	Green	The power supply, #1, is working normally.
	Amber	Power is present, but not ON. Power is faulty.
	Off	Power supply is not present.

Port LED Definition for Edgecore AGR400

The network port LEDs are used to represent the status of the link associated with the LED.

Port type	Condition	Status
SFP28	Up	Blue LED is on (Single LED per port)
	Down	LED is Off
QSFP28	Up	Blue LED is on (2 LEDs per port)
	Down	LED is Off
QSFP-DD	Up	Not tested
	Down	Not tested

System LED Definition for Edgecore AGR400

The system LEDs are used to indicate the status of power and system.

LED	Condition	Status
DIAG (Diagnostic)	Green	System self diagnostic test is successfully completed.
	Amber	System self-diagnostic test has detected a fault. (Faulty Fan, thermal, or any interface.)
LOC	Flashing Amber	Flashing by remote management command. Assists the technician in finding the right device for service in the rack.
	Off	Not the switch that the technician needs to find.
FAN	Green	System fan is working normally.
	Amber	Fan tray is present but system fan is faulty.
	Off	System is off.

LED	Condition	Status
PSU0 (power supply status)	Green	The power supply, #0, is working normally.
	Amber	Power is present, but not ON. Power is faulty.
	Off	Power supply is not present.
PSU1 (power supply status)	Green	The power supply, #1, is working normally.
	Amber	Power is present, but not ON. Power is faulty.
	Off	Power supply is not present.

Port LED Definition for Edgecore CSR320

The network port LEDs are used to represent the status of the link associated with the LED.

Port type	Condition	Status
SFP+	Up	Green LED is on
	Down	LED is Off
SFP28	Up	Green LED is on
	Down	LED is Off
QSFP28	Up	Green LED is on
	Down	LED is off

System LED Definition for Edgecore CSR320

The system LEDs are used to indicate the status of power and system.

Port type	Condition	Status
PSU0 (power supply status)	Green	The power supply, #0, is working normally.
	Amber	Power is present, but not ON. Power supply is faulty.
	Off	Power supply is not present.
PSU1 (power supply status)	Green	The power supply, #1, is working normally.
	Amber	Power is present, but not ON. Power supply is faulty.
	Off	Power supply is not present.
DIAG (Diagnostic)	Green	System self diagnostic test is successfully completed.
	Blinking Green	System self-diagnostic test is in progress
	Amber	System self-diagnostic test has detected a fault.
FAN	Green	System fan is working normally.
	Blinking Green	System FAN tray is powered off when ambient temperature is less than 10 degree C.
	Amber	Fan tray is present but faulty.
LOC	Flashing Amber	Flashing by remote management command. Assists the technician in finding the right device for service in the rack.
	Off	Not the switch that the technician needs to find.

Port LED Definition for Delta AGCVA48S

The network port LEDs are used to show the status of the link associated with the LED. In the Delta’s AGCVA48S switch, there are 4*10G SFP+ ports and 48*25G

SFP28 ports, and 10*100G QSFP28 ports on the front panel. The SFP+, SFP28, and QSFP28 ports operate in full duplex mode even when the speed is 10Gbps, 25Gbps and 100Gbps respectively.

Port type	Condition	Status
SFP+ Port LED	On (Green)	Link is up
	Off	No link
SFP28 Port LED	On (Green)	Link is up
	Off	No link
QSFP28 Port LED	On (Green)	Link is up
	Off	No link

Port LED Definition for Edgecore CSR440

The network port LEDs are used to represent the status of the link associated with the LED.

Port type	Condition	Status
10G	Up	Green
	Down	LED is Off
25G	Up	Blue
	Down	LED is Off
100G	Up	Two LEDs per port. Green (on QSFP28) and Violet (on QSFP-DD).
	Down	LED is off

System LED Definition for Edgecore CSR440

The system LEDs are used to indicate the status of power and system.

Port type	Condition	Status
PSU1 (power supply unit)	Solid Green	PSU1 is operating normally.
	Solid Amber	Input power is present, but the PSU1 is not ON or is in a faulty state.
	Off	The input power is not present or PSU1 is not present.
PSU2 (power supply unit)	Solid Green	PSU2 is operating normally.
	Solid Amber	Input power is present, but the PSU2 is not ON or is in a faulty state.
	Off	Input power is not present or PSU2 is not present.
DIAG (Diagnostic)	Solid Green	System self diagnostic test is successfully completed.
	Blinking Green	System self-diagnostic test is in progress.
	Solid Orange	System self-diagnostic test has detected a fault.
FAN	Solid Green	All fan modules are present and working normally.
	Solid Orange	Fan module is present but a fault has been detected.
	Off	Fan module is not present.
LOC	Blinking Blue	Flashing by remote management command. It helps technicians in finding the right device for service in the rack. It provides an indication of the location of the device.
	Off	No indication.
ALARM	Solid Green	No alarms detected.
	Solid Red	Local alarm is detected.

15.4. IPMI

15.4.1. IPMI Overview

The IPMI (Intelligent Platform Management Interface) is an open-standard hardware management interface that allows you communicate with BMC (Baseboard Management Controller) of the platform hardware.

BMC, a dedicated micro-controller, manages the interface between system-management software and platform hardware. BMC performs operations such as remote power on or power off, or makes the console accessible. BMC also monitors the hardware resources such as sensors and can send alert messages over the LAN indicating a potential failure of the system.

IPMI is a way to manage the platform hardware by interacting with the hardware directly rather than with the operating system. The advantages of using IPMI are that it allows an out-of-band platform hardware management and the operating system is not burdened with the operational overheads by sending system status data.

Supported Platforms

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

15.4.2. IPMI Configuration

Configuration Syntax and Commands

The following sections describe the IPMI configuration syntax and commands.

IPMI Interface Configuration

This configuration allows you to communicate with the BMC on the platform hardware.

Syntax:

set system platform-management ipmi interface <interface-id> <attribute>

<value>

Attribute	Description
interface <interface-id>	IPMI channel ID. For UfiSpace switches, the LAN interface ID is 1.
type <lan>	type of network
mode <dhcp|static>	Manually set the static IPv4 address or configure DHCP to receive dynamic IPv4 address.
address-family ipv4 prefix4 <prefix4>	IPv4 prefix
address-family ipv4 gateway-address <gateway-address>	IPv4 address for gateway

Example: IPMI Interface Configuration

```
{
  "ietf-restconf:data": {
    "rtbrick-config:system": {
      "platform-management": {
        "ipmi": {
          "interface": [
            {
              "id": 1,
              "type": "lan",
              "mode": "static",
              "address-family": {
                "ipv4": {
                  "prefix4": "198.51.100.110/24",
                  "gateway-address": "198.51.100.111"
                }
              }
            }
          ]
        }
      }
    }
  }
}
```

IPMI User Configuration

This configuration allows you to change the password of one of the 10 users, including the admin user.

Syntax:

set system platform-management ipmi user <user-id> <options>

Attribute	Description
user <user-id>	User ID. The ipmi user id of the default 'admin' user on UfiSpace switches is 2.
password-plain-text <password-plain-text>	Plain text password
password-encrypted-text <password-encrypted-text>	Encrypted password

Example: IPMI User Configuration

```
{
  "ietf-restconf:data": {
    "rtbrick-config:system": {
      "platform-management": {
        "ipmi": {
          "user": [
            {
              "id": 2,
              "password-encrypted-text": "$21e4946e31b406de98b3077aef03ed5a7"
            }
          ]
        }
      }
    }
  }
}
```

15.4.3. IPMI Operational Commands

IPMI Validation Commands

The IPMI validation commands provide you the IPMI configuration, channel, and user information. Configure the server that connects to the hardware platform. Use the IPMI tool to verify the remote accessibility by the running the following commands on the Linux shell.

Verification of remote access can be performed using the following command:

ipmitool -I lanplus -H <ip address> -U admin -P admin lan print 1

Example for remote access verification

```
supervisor@srv10-tst:~$ ipmitool -I lanplus -H 198.51.100.100 -U admin -P admin
lan print 1
Set in Progress         : Set Complete
Auth Type Support       :
Auth Type Enable        : Callback : MD5
                        : User     : MD5
                        : Operator : MD5
                        : Admin    : MD5
                        : OEM      : MD5
IP Address Source       : Static Address
IP Address              : 198.51.100.100
Subnet Mask             : 255.255.255.128
MAC Address             : e8:c5:7a:8f:78:0d
SNMP Community String   : AMI
IP Header               : TTL=0x40 Flags=0x40 Precedence=0x00 TOS=0x10
BMC ARP Control         : ARP Responses Enabled, Gratuitous ARP Disabled
Gratituous ARP Intrvl   : 0.0 seconds
Default Gateway IP      : 198.51.100.41
Default Gateway MAC     : 00:00:5e:00:01:01
Backup Gateway IP       : 198.51.100.10
Backup Gateway MAC      : 00:00:00:00:00:00
802.1q VLAN ID          : Disabled
802.1q VLAN Priority    : 0
RMCP+ Cipher Suites     : 0,1,2,3,6,7,8,11,12,15,16,17
Cipher Suite Priv Max   : caaaaaaaaaaaXXX
                        :     X=Cipher Suite Unused
                        :     c=CALLBACK
                        :     u=USER
                        :     o=OPERATOR
                        :     a=ADMIN
                        :     O=OEM
Bad Password Threshold  : 0
Invalid password disable: no
Attempt Count Reset Int.: 0
User Lockout Interval   : 0
```

Use the following command to verify the channel information:

sudo ipmitool channel info 1

Example for channel information

```
supervisor@onl>ufi06.q2c.u25.r4.nbg.rtbrick.net:~ $ sudo ipmitool channel info 1
Channel 0x1 info:
  Channel Medium Type   : 802.3 LAN
  Channel Protocol Type : IPMB-1.0
  Session Support       : multi-session
  Active Session Count  : 0
  Protocol Vendor ID    : 7154
  Volatile(active) Settings
    Alerting            : enabled
    Per-message Auth    : disabled
    User Level Auth     : enabled
    Access Mode         : always available
  Non-Volatile Settings
    Alerting            : enabled
```

```
     Per-message Auth     : disabled
     User Level Auth      : enabled
     Access Mode          : always available
```

Use the following command to verify a specific channel information:

sudo ipmitool lan print 1

Example for specific channel information

```
supervisor@onl>ufi06.q2c.u25.r4.nbg.rtbrick.net:~ $ sudo ipmitool lan print 1
Set in Progress         : Set Complete
Auth Type Support       :
Auth Type Enable        : Callback : MD5
                        : User     : MD5
                        : Operator : MD5
                        : Admin    : MD5
                        : OEM      : MD5
IP Address Source       : Static Address
IP Address              : 198.51.100.100
Subnet Mask             : 255.255.255.128
MAC Address             : e8:c5:7a:8f:78:0d
SNMP Community String   : AMI
IP Header               : TTL=0x40 Flags=0x40 Precedence=0x00 TOS=0x10
BMC ARP Control         : ARP Responses Enabled, Gratuitous ARP Disabled
Gratituous ARP Intrvl   : 0.0 seconds
Default Gateway IP      : 198.51.100.41
Default Gateway MAC     : 00:00:5e:00:01:01
Backup Gateway IP       : 198.51.100.10
Backup Gateway MAC      : 00:00:00:00:00:00
802.1q VLAN ID          : Disabled
802.1q VLAN Priority    : 0
RMCP+ Cipher Suites     : 0,1,2,3,6,7,8,11,12,15,16,17
Cipher Suite Priv Max   : caaaaaaaaaaaXXX
                        :     X=Cipher Suite Unused
                        :     c=CALLBACK
                        :     u=USER
                        :     o=OPERATOR
                        :     a=ADMIN
                        :     O=OEM
Bad Password Threshold  : 0
Invalid password disable: no
Attempt Count Reset Int.: 0
User Lockout Interval   : 0
```

Use the following command to verify the user for a channel:

sudo ipmitool user list 1

Example for user per channel

```
supervisor@onl>ufi06.q2c.u25.r4.nbg.rtbrick.net:~ $ sudo ipmitool user list 1
ID  Name      Callin  Link Auth IPMI Msg   Channel Priv Limit
```

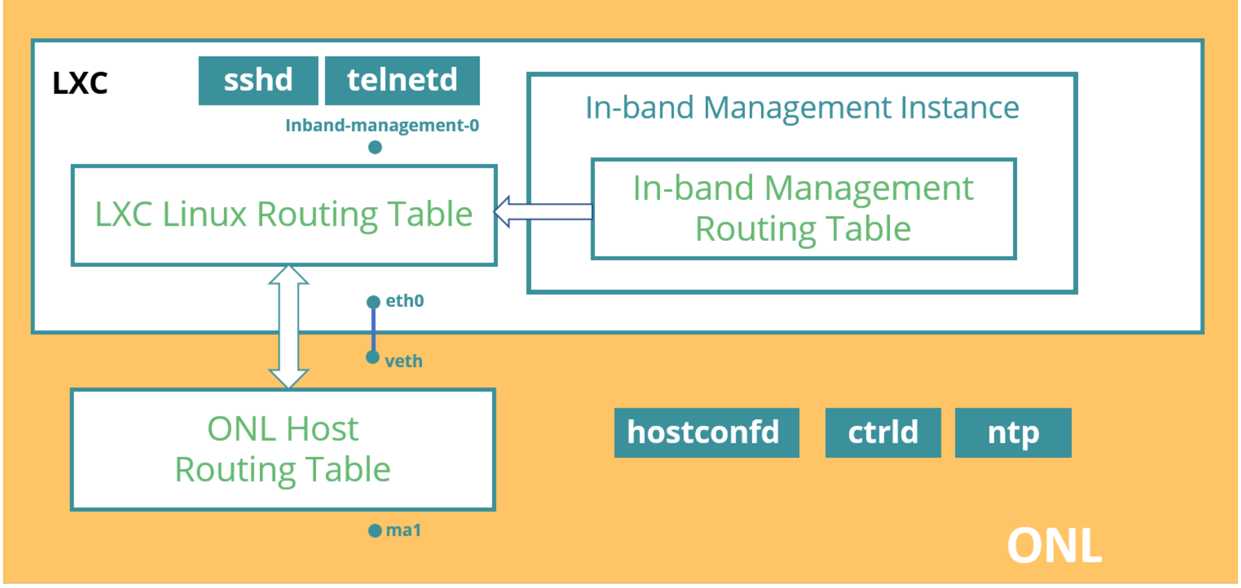
```
1                    false   false      true       ADMINISTRATOR
2   admin            false   false      true       ADMINISTRATOR
3                    true    false      false      NO ACCESS
4                    true    false      false      NO ACCESS
5                    true    false      false      NO ACCESS
6                    true    false      false      NO ACCESS
7                    true    false      false      NO ACCESS
8                    true    false      false      NO ACCESS
9                    true    false      false      NO ACCESS
10                   true    false      false      NO ACCESS
```

15.5. Inband Management

15.5.1. In-band Management Overview

RBFS is mostly deployed on an ONL host as a Linux container. The ONL host is only reachable through the out-of-band management interface. In order to use services like NTP, and TACACS, which are run on ONL, or to use services like ssh, snmpd running in LXC container, one must use an out-of-band management connection. In-band management provides a way to access these services which are running in ONL and LXC containers via physical traffic ports.

The RBFS creates a Linux kernel interface named inband-mgmt-0 when in-band management is enabled on an instance. The loopback IPs of the in-band instance are then assigned to this Linux interface, and the routes of this instance are downloaded to the LXC container, then to ONL. Trap rules are installed in the hardware depending on the in-band service enabled.



Supported Platforms

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

15.5.2. In-band Management Configuration

Enabling In-band Management in an Instance

Use the following CLI syntax to enable in-band management in an instance:

set inband management instance <instance-name>

Attribute	Description
<instance-name>	Routing instance name in which in-band management has to be enabled. All the IFLs in this instance will be enabled with in-band management service after executing this command

The following example configures the management instance in which in-band management will be enabled:

```
set inband management instance management
```

The following example shows in-band management in an instance:

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

Enabling In-band Management Services

Syntax:

set inband management instance <instance-name> <service> <true/false>

Attribute	Description
<instance-name>	Routing instance name in which in-band management has to be enabled. All the IFLs in this instance will be enabled with in-band management service after executing this command
<service>	Specifies the supported services to enable: apigw, ctrld, ntp, snmp, ssh, tacacs, telnet. By enabling any of these services, hosts reachable via the physical interface in the inband instance can access the services.
<true | false>	A true value enables the service. A false value disables the service.

Example: Enabling In-band Management Services

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

Enabling API Gateway (APIGW) Service

To access the APIGW service running in the ONL, this service has to be enabled in in-band management.

Syntax:

set inband management instance <instance-name> **apigw** <true/false>

Attribute	Description
<instance-name>	Routing instance name in which in-band management has to be enabled.
<true | false>	A true value enables the APIGW service. A false value disables the APIGW service.

Example: Enabling APIGW In-band Management Services

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

Enabling CTRLD Service

To access the CTRLD service running in the ONL, the CTRLD service has to be enabled in in-band management.

Syntax:

set inband management instance <instance-name> **ctrld** <true/false>

Attribute	Description
<instance-name>	Routing instance name in which in-band management has to be enabled.
<true | false>	A true value enables the CTRLD service. A false value disables the CTRLD service.

Example: Enabling CTRLD In-band Management Services

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

Enabling NTP service

To access the NTP service running in the ONL, this service has to be enabled in in-band management.

Syntax:

set inband management instance <instance-name> **ntp** <true/false>

Attribute	Description
<instance-name>	Routing instance name in which in-band management has to be enabled.
<true | false>	A true value enables the ntp service. A false value disables the ntp service.

Example: Enabling NTP In-band Management Services

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

Enabling SNMP service

To access the Simple Network Management Protocol (SNMP) service running in the ONL, this service has to be enabled in in-band management.

Syntax:

set inband management instance <instance-name> **snmp** <true/false>

Attribute	Description
<instance-name>	routing instance name in which in-band management has to be enabled.
<true | false>	A true value enables the SNMP service. A false value disables the SNMP service.

Example: Enabling SNMP In-band Management Services

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

Enabling SSH service

To access the ssh service running in the LXC container hosting RBFS, ssh service has to be enabled.

Syntax:

set inband management instance <instance-name> **ssh** <true/false>

Attribute	Description
<instance-name>	Routing instance name in which in-band management has to be enabled.
<true | false>	A true value enables the ssh service. A false value disables the ssh service.

Example: Enabling SSH In-band Management Services

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

Enabling TACACS Service

To access the TACACS service running in the ONL, this service has to be enabled in in-band management.

Syntax:

set inband management instance <instance-name> **tacacs** <true/false>

Attribute	Description
<instance-name>	Routing instance name in which in-band management has to be enabled.
<true | false>	A true value enables the TACACS service. A false value disables the TACACS service.

Example: Enabling TACACS In-band Management Services

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

Enabling Telnet Service

To access the telnet service running in the LXC container hosting RBFS, telnet service has to be enabled.

Syntax:

set inband management instance <instance-name> **telnet** <true/false>

Attribute	Description
<instance-name>	Routing instance name in which in-band management has to be enabled.
<true | false>	A true value enables the telnet service. A false value disables the telnet service.

Example: Enabling Telnet In-band Management Services

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

Enabling Connection Tracking

Enabling connection tracking in inband installs dynamic ACLs for all the connection/packet initiated by the device so that the response packets are not dropped at the hardware.

Syntax:

set inband management instance <instance-name> **connection-tracking true**

Attribute	Description
<instance-name>	Routing instance name in which in-band management has to be enabled.
true	Enables all in-band management services.

Example: Enabling Connection Tracking in In-band Management

```
"rtbrick-config:inband-management": {
    "instance": [
      {
        "name": "default",
        "connection-tracking": "true"
      }
    ]
  }
```

Enabling All Services in In-band Management

Enabling this service will allow access to all services running in LXC/ONL. Once this service is enabled, packets that don’t hit any of the other acls/services in RBFS are redirected to LXC/ONL.

Syntax:

set inband management instance <instance-name> **all true**

Attribute	Description
<instance-name>	Routing instance name in which in-band management has to be enabled.
true	Enables all in-band management services.

Example: Enabling all In-band Management Services

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

Enabling In-band Management for a Specific Source

Enabling any of the in-band services as mentioned in the previous section will expose this service to all the sources which are reachable via in-band service.

To restrict this to specific source prefixes, source-prefix-list has to be enabled using the following command.

By configuring this, the hosts having IPs in the mentioned source prefix list only can access this service.

Syntax:

set inband management instance <instance-name> **source-prefix-list** <source-prefix-list-name>

Attribute	Description
<instance-name>	Routing instance name in which in-band management has to be enabled.
<source-prefix-list-name>	Specifies the name of the source prefix-list which is configured in 'set forwarding-options prefix-list' command.

Example: Enabling source prefix list in In-band Management Services

```
    "rtbrick-config:inband-management": {
      "instance": [
        {
          "name": "management",
          "source-prefix-list": "source-prefix1"
        }
      ]
    }
```

15.5.3. In-Band Management Operational Commands

The In-band Management show commands provide detailed information about the In-band Management operations.

Verifying In-band Management on LXC Container

In the LXC container, there will be a new interface named inband-mgmt-0 on enabling in-band management. All the loopback address as well as route in in-band instance should be assigned to this interface.

The example below shows how to verify if inband-mgmt-0 interface is created and if the routing for management traffic is pointing to it.

```
supervisor@rtbrick:~$ ip link show
<...>
5: inband-mgmt-0: <POINTOPOINT,UP,LOWER_UP> mtu 1492 qdisc pfifo_fast state
UNKNOWN mode DEFAULT group default qlen 500
    link/none

supervisor@rtbrick:~$ ip route show
198.51.100.1/24 dev inband-mgmt-0 proto rtb_fibd scope link
<...>
```

Verifying In-band Management on ONL

The example below shows how to verify Linux routing tables on ONL host and LXC container.

```
root@bl2-pod1:~# ip route show
default via 198.51.100.202 dev ma1 proto rtb_routesync metric 4294966272
default via 198.51.100.10 dev lxcbr0 proto rtb_routesync scope rtb_umc metric 128
198.51.100.81/24 via 198.51.100.10 dev lxcbr0 proto rtb_routesync scope rtb_umc
metric 128
198.51.100.30/24 dev lxcbr0 proto kernel scope link src 198.51.100.31
198.51.100.55 via 198.51.100.10 dev lxcbr0 proto rtb_routesync scope rtb_umc
metric 128
198.51.100.119/23 dev ma1 proto kernel scope link src 198.51.100.112

supervisor@rtbrick:~$ ip route show
default via 198.51.100.31 dev eth0 proto rtb_routesync scope rtb_umc metric
4294966400
default dev inband-mgmt-0 proto rtb_fibd scope link
198.51.100.81/24 dev inband-mgmt-0 proto rtb_fibd scope link
198.51.100.30/24 dev eth0 proto kernel scope link src 198.51.100.10
198.51.100.55 dev inband-mgmt-0 proto rtb_fibd scope link
198.51.100.119/23 via 198.51.100.31 dev eth0 proto rtb_routesync scope rtb_umc
metric 128
198.51.100.112 via 198.51.100.31 dev eth0 proto rtb_routesync scope rtb_umc metric
128
```

15.6. DHCP Relay/Proxy

15.6.1. DHCP Relay Overview

The Dynamic Host Configuration Protocol (DHCP) is a standardized client/server network protocol that dynamically assigns IPv4 addresses and other related configuration information to network devices.

DHCP provides an automated way to distribute and update IPv4 addresses and other configuration information on a network. A DHCP server provides this information to a DHCP client through the exchange of a series of messages, known as the DHCP conversation or the DHCP transaction. If the DHCP server and DHCP clients are located on different subnets, a DHCP relay agent is used to facilitate the conversation.

The RBFS DHCP relay agent handles all DHCP packets received on logical interfaces with the relay agent functionality enabled. This includes all DHCP broadcast and unicast packets to ensure that the whole DHCP communication between client and server can be tracked for operational and security reasons. The DHCP packets received from clients will than be forwarded to one or more DHCP servers to support high availability.

All DHCP packets send from the relay agent to the configured DHCP servers will be send with the IP address of the gateway interface (giaddr) or a dedicated source address per server. This allows to better traverse firewalls between relay agent and server.

Supported Platforms

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

DHCP Relay Agent

The following are the operation steps involved by the DHCP relay agent with multiple clients and servers:

1. DHCP client sends DISCOVER packets onto the LAN.
2. DHCP relay agent receives the DISCOVER packet, creates a client binding and forwards it to the configured server(s) in different subnets.
3. DHCP server replies with OFFER packet.
4. DHCP relay agent on receiving the OFFER, sends it to the client based on the binding.

5. DHCP client sends a REQUEST after receiving the OFFER.
6. DHCP relay agent forwards the REQUEST to the configured server(s).
7. Selected DHCP server replies to the REQUEST with an ACK.
8. DHCP relay agent receives the ACK and forward it to client.
9. Client will start using the IP and configuration after receiving the ACK.
10. DHCP relay agent listen to the unicast communication between the client and server and update the bindings.

DHCP Relay in Proxy Mode

The proxy mode is an enhancement for the DHCP relay function to hide and protect the actual DHCP servers from the clients. For servers this mode is transparent but for clients it seems that the relay agent is the server and the actual servers are not visible anymore.

The following are the operation steps involved by the DHCP relay agent in proxy mode with multiple clients and servers:

1. DHCP client sends DISCOVER packets onto the LAN.
2. DHCP relay agent receives the DISCOVER packet, creates a client binding and forwards it to the configured server(s) in different subnets.
3. DHCP server replies with OFFER packet.
4. DHCP relay agent on receiving OFFERs from multiple servers, select the OFFER from the first server, replaces the server identifier with relay agent address, update the binding and sends it to the client.
5. DHCP client sends a REQUEST to relay agent address after receiving the OFFER.
6. DHCP relay agent forwards the REQUEST to the configured server(s).
7. Selected DHCP server replies to the REQUEST with an ACK.
8. DHCP relay agent receives the ACK, replaces the server identifier with relay agent address and forward it to client.
9. Client will start using the IP and configuration after receiving the ACK.
10. Client request the relay agent for the lease renewals, which relay agent will forward to the selected server.

In all the steps above, the client remains unaware of the actual DHCP server.

15.6.2. DHCP Relay Configuration

DHCP relay requires a pre-provisioned logical interface with at least an local or borrowed (unnumbered) IPv4 address configured which is than used a gateway IP address (giaddr) by the DHCP relay function.

There are two steps required to enable the DHCP relay functionality:

1. DHCP server configuration
2. DHCP relay configuration

DHCP Server Configuration

This configuration defines the DHCP servers which are referenced by the actual DHCP relay configuration.

```
supervisor@rtbrick>LEAF01: cfg> set access dhcp-server <server>
  <cr>
  address                 DHCP server address
  routing-instance        Instance name from which DHCP server is reachable
  source-address          Source address used for DHCP packets
```

The following example shows a typical DHCP server configuration.

```
{
  "rtbrick-config:access": {
    "dhcp-server": [
      {
        "server-name": "server1",
        "address": "198.51.100.101",
        "routing-instance": "default"
      }
    ]
  }
}
```

Attribute	Description
server	Specifies the DHCP server name.
address	Specifies the IPv4 address of the DHCP server.
routing-instance	Routing instance from which DHCP server is reachable. **Default:** default

Attribute	Description
source-address <source-address>	Specifies the source IPv4 address to be used to reach DHCP server. **Default:** gateway interfaces address (giaddr)

DHCP Relay Configuration

This configuration enables the DHCP relay function on the corresponding logical interface (IFL).

```
supervisor@rtbrick>LEAF01: cfg> set access dhcp-relay <interface-name>
  <cr>
  agent-circuit-id        Add Agent-Circuit-Id (option 82)
  agent-remote-id         Add Agent-Remote-ID (option 82)
  dhcp-server             DHCP server
  proxy-mode              Enable relay proxy mode
```

The following example shows a typical DHCP relay configuration.

```
{
    "rtbrick-config:access": {
      "dhcp-relay": [
        {
          "interface": "ifl-0/0/1/1",
          "dhcp-server": [
            "server1",
            "server2"
           ]
        }
      ]
    }
  }
```

Attribute	Description
interface-name	Logical interface name on which client packets are expected.
dhcp-server <dhcp-server>	DHCP server name to which client packets has to be forwarded (multiple servers can be configured).
proxy-mode	Enable the relay proxy mode for this interface.
agent-circuit-id <aci>	Agent-Circuit-Id to be added in DHCP option 82.
agent-remote-id <ari>	Agent-Remote-Id to be added in DHCP option 82.

DHCP Relay on Unnumbered Interface

DHCP Relay is supported over unnumbered interfaces. The configuration is similar to the regular DHCP relay configuration. The only change is that the logical interface will have a borrowed IP address.

The following example shows a typical DHCP relay configuration over unnumbered interface.

```
{
  "data": {
    "rtbrick-config:interface": [
      {
        "name": "ifp-0/0/1",
        "unit": [
          {
            "unit-id": 1,
            "unnumbered": {
              "interface": "lo-0/0/1/1"
            }
          }
        ]
      },
      {
        "name": "ifp-0/0/2",
        "unit": [
          {
            "unit-id": 1,
            "unnumbered": {
              "interface": "lo-0/0/1/1"
            }
          }
        ]
      },
      {
        "name": "lo-0/0/1",
        "unit": [
          {
            "unit-id": 1,
            "address": {
              "ipv4": [
                {
                  "prefix4": "198.51.100.71/24"
                }
              ]
            }
          }
        ]
      }
    ],
    "rtbrick-config:access": {
      "dhcp-relay": [
        {
          "interface": "ifl-0/0/1/1",
          "dhcp-server": [
            "server1",
            "server2"
            ]
        },
        {
          "interface": "ifl-0/0/2/1",
          "dhcp-server": [
            "server1",
            "server2"
            ]
        }
      ]
    }
  }
}
```

15.6.3. DHCP Relay Operational Commands

Verify DHCP Relay Configuration

A good starting point for any troubleshooting is to get an overview about the configuration.

First check the DHCP server configuration and optionally check the reachability using additional commands like ping.

```
supervisor@rtbrick>LEAF01: cfg> show config access dhcp-server
{
    "rtbrick-config:dhcp-server": [
      {
        "server-name": "server1",
        "address": "198.51.100.15",
        "source-address": "198.51.100.101",
        "routing-instance": "default"
      }
    ]
  }
```

In the next step it should be verified that the DHCP relay function is enabled for the desired logical interfaces. Those interfaces must be configured with a valid local or borrowed (unnumbered) IPv4 address.

```
supervisor@rtbrick>LEAF01: cfg> show config access dhcp-relay
{
    "rtbrick-config:dhcp-relay": [
      {
        "interface": "ifl-0/0/1/1",
        "dhcp-server": [
          "server1",
          "server2"
          ]
      }
    ]
  }
```

Verify DHCP Relay Packet Processing

All DHCP packets processed by RBFS will be handled by the IPoE daemon (rtbrick-ipoed.1) which separates between DHCP relay and subscriber packets.

The command show dhcp statistics gives some statistics about the DHCP packets send and received by the IPoE daemon.

```
supervisor@rtbrick>LEAF01: op> show dhcp statistics
Packets received            : 0
Decode error                : 0
Relay packets received      : 0
Relay packets sent          : 0
Relay send error            : 0
Subscriber packets received : 0
Subscriber packets sent     : 0
Subscriber send error       : 0
```

Received malformed packets will be counted here as part of the shared DHCP packet infrastructure.

Those statistics can be reset with the command clear dhcp statistics.

In the next step, the command show dhcp relay statistics provides more detailed statistics for those packets handled by the DHCP relay function.

```
supervisor@rtbrick>LEAF01: op> show dhcp relay statistics
Packet           Received         Sent
DHCP DISCOVER  : 0                0
DHCP OFFER     : 0                0
DHCP REQUEST   : 0                0
DHCP DECLINE   : 0                0
DHCP ACK       : 0                0
DHCP NAK       : 0                0
DHCP RELEASE   : 0                0
DHCP INFORM    : 0                0

Errors:

Invalid client packets received : 0
Invalid server packets received : 0
I/O errors                      : 0
Configuration errors            : 0
```

A DHCP packet is consider as invalid if correctly formed but some invalid or unexpected values like a DHCP hop count greater than 16, messages from server without server identifier option, non-matching transaction (XID) and many more.

Those statistics can be reset with the command clear dhcp relay statistics.

Verify DHCP Relay Bindings

The DHCP relay function tracks all active DHCP communications as so called bindings. The command show dhcp relay binding allows to verify those bindings with extensive filters to better navigate in scaled environments with thousands of bindings.

```
supervisor@rtbrick>LEAF01: op> show dhcp relay binding ?
  <cr>
  detail                    Detailed binding information
  filter                    Filter DHCP relay binding
  interface                 Interface
```

```
supervisor@rtbrick>LEAF01: op> show dhcp relay binding filter ?
  address                   Client IP address
  count                     Count matching bindings
  detail                    Detailed binding information
  interface                 Interface
  interface-regex           Interface regex
  mac                       Hardware address
  mac-regex                 Hardware address regex
  server                    Server identifier
```

```
supervisor@rtbrick>LEAF01: op> show dhcp relay binding detail
Interface                    : ifp-0/0/1/3
Hardware Address             : 02:00:00:00:00:01
Client IP Address            : 198.51.100.58
Routing Instance             : default
Proxy Mode                   : enabled
Server Identifier            : 198.51.100.15
Gateway IP Address           : 198.51.100.35
Lease Time                   : 120.0 sec
Lease Start                  : Tue Nov 16 22:27:15 GMT +0000 2021
Lease Expires                : 66.0 sec
Last Packet Received         : Tue Nov 16 22:27:15 GMT +0000 2021
Last Received Packet Type    : ACK
Last Packet Sent             : Tue Nov 16 22:27:15 GMT +0000 2021
Last Sent Packet Type        : ACK
```

The most extensive output is provided by selecting a particular binding using the interface and client MAC address as key.

```
supervisor@rtbrick>LEAF01: op> show dhcp relay binding interface ifp-0/0/1/3 mac 02:00:00:00:00:01
```

```
Interface                  : ifp-0/0/1/3
Hardware Address           : 02:00:00:00:00:01
Client IP Address          : 198.51.100.58
Routing Instance           : default
Proxy Mode                 : enabled
Server Identifier          : 198.51.100.15
Gateway IP Address         : 198.51.100.35
Lease Time                 : 120.0 sec
Lease Start                : Tue Nov 16 22:31:15 GMT +0000 2021
Lease Expires              : 104.0 sec
Last Packet Received       : Tue Nov 16 22:31:15 GMT +0000 2021
Last Received Packet Type  : ACK
Last Packet Sent           : Tue Nov 16 22:31:15 GMT +0000 2021
Last Sent Packet Type      : ACK

Statistics:

Packet           Received         Sent
DHCP DISCOVER  : 1                1
DHCP OFFER     : 1                1
DHCP REQUEST   : 36               36
DHCP DECLINE   : 0                0
DHCP ACK       : 36               36
DHCP NAK       : 0                0
DHCP RELEASE   : 0                0
DHCP INFORM    : 0                0

Errors:

Invalid client packets received : 0
Invalid server packets received : 0
I/O errors                      : 0
Configuration errors            : 0
```

This output returns additional statistics per DHCP relay binding.

The command clear dhcp relay binding removes either all bindings, all bindings of a corresponding interface or only one particular binding.

```
supervisor@rtbrick>LEAF01: op> clear dhcp relay binding ?
  all                     Clear all DHCP relay bindings
  interface               Interface
```

```
supervisor@rtbrick>LEAF01: op> clear dhcp relay binding all
supervisor@rtbrick>LEAF01: op> clear dhcp relay binding interface ifp-0/0/1/3
supervisor@rtbrick>LEAF01: op> clear dhcp relay binding interface ifp-0/0/1/3 mac
02:00:00:00:00:01
```

16. Telemetry

16.1. TSDB

16.1.1. Prometheus Time Series Database (TSDB) Integration Overview

Operational-state visibility is key for troubleshooting, testing, monitoring and capacity management. This requires to sample router metrics periodically. Ingestion of time-series data allows to ask of interesting operational queries.

Examples:

- A slightly increasing memory consumption over time while the overall PPPoE session count has not changed, for example, is an indication of a memory leak.
- If the 5-minute chassis temperature is too high, this might be an indication for an imminent hardware breakdown and the switch hardware must be replaced.
- If utilization of all fabric interfaces is constantly touching the 80% saturation levels then new fabric links must be commissioned.
- High input traffic with degradation of optical receive levels might be an indication of running very close to the optical budget.

The challenge is to sample all this information efficiently in terms of disk, memory and CPU utilization while providing comprehensive query and reporting functionality.

Supported Platforms

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

Architectural Overview

The RBFS telemetry architecture is based on Prometheus as an open-source systems monitoring and alerting toolkit. Prometheus is designed to pull metrics periodically and save them efficiently. It allows us to analyze the metrics with a powerful query language called PromQL. Also, an optional alert management is available. There is an opportunity to tie it together with its own services to integrate it into the system landscape. Data should have short retention times (default 15d).

This fits perfectly with the needs of BDS. The figure below shows how it fits in an overall architecture.

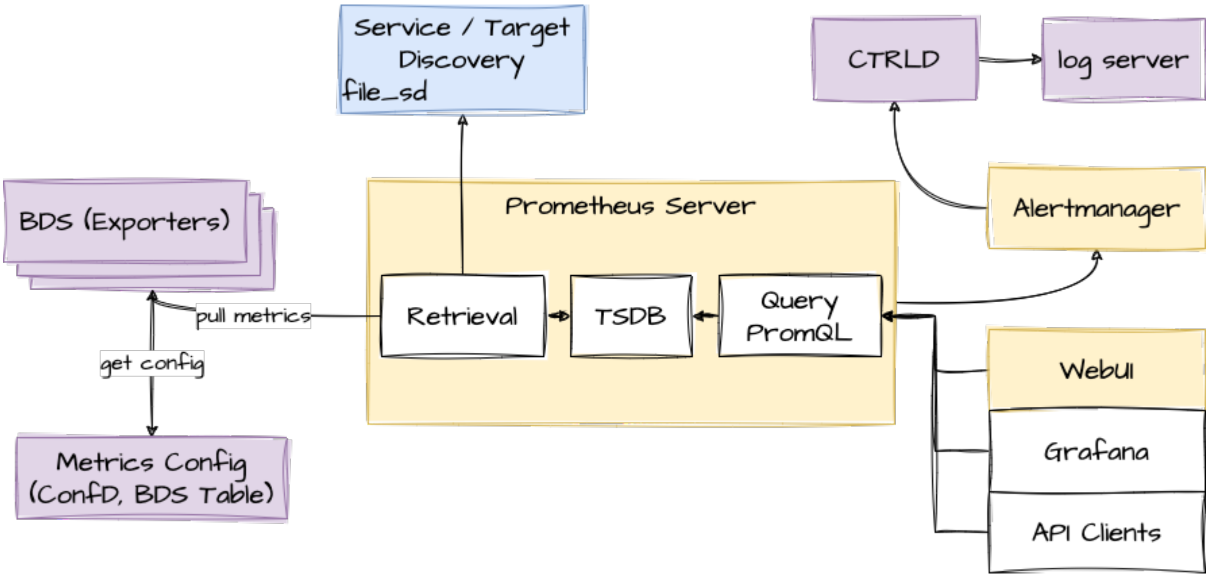


Figure 27. Prometheus in RBFS

To mitigate the short retention times, which fits to BDS but not in an overall telemetry process, the data can be stored in a centralized storage database (for example, Influx) this can be done by federation or via remote storage adapters. To distribute the alert messages from Prometheus, CTRLD functions as an "alertmanager webhook receiver", which takes the alert and distributes it to a log management tool.

Router deployment model

Prometheus DB is run on the router as a dedicated process. It ships with a package-time configuration to poll each BDS-capable speaker at periodic intervals. Initially the periodic interval is 1 second. The Prometheus Exposition format is a very simple HTTP-based GET query that asks a given BD speaker "Give me all your metrics". Each BD subscribes to the *global.time-series.metric.config* table, which contains an operator-configurable list of BDS targets. Only the BDS which is the master of a table responds. Next Prometheus polls the BD using the */metrics* URL.

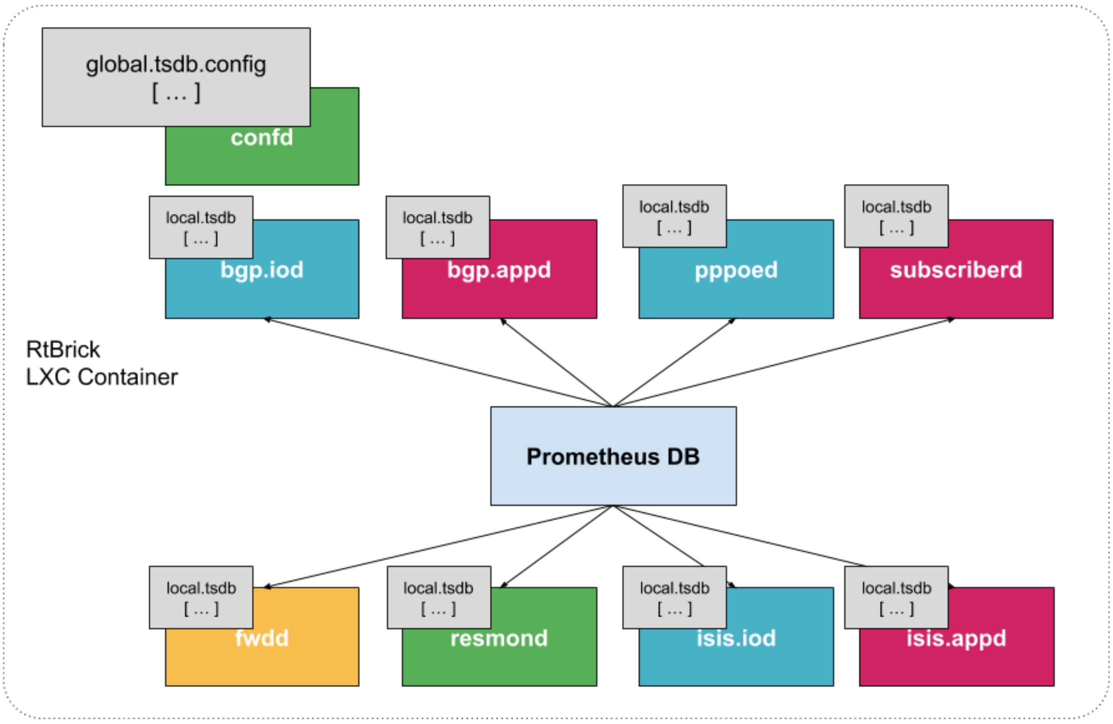


Figure 28. Prometheus in RBFS with the different scrape target

Storage efficiency

On average Prometheus uses only around 1-2 bytes per sample. Thus, to plan the capacity of a Prometheus server, you can use the rough formula:

```
needed_disk_space = retention_time_seconds * ingested_samples_per_second *
bytes_per_sample
```

The single binaries disk space:

```
-rwxr-xr-x 1 root root 27M Sep 2 22:51 alertmanager +
-rwxr-xr-x 1 root root 81M Sep 2 22:51 prometheus +
-rwxr-xr-x 1 root root 49M Sep 3 19:55 promtool
```

Promtool is needed to test the configurations before set them to prometheus.

Alerting

The alerting is configured through Prometheus. For more information, see alertmanager.

Role of CTRLD

Prometheus and Alertmanager register themself in CTRLD, so that CTRLD is aware of these two services.

Refer to Figure-1, which provides an overview of the role of CTRLD.

Service state and Proxy

The registration of the services gives 2 advantages:

1. The operational state is an indicator that the service is up and running.
2. The proxy functionality of CTRLD can be used for Prometheus and alertmanager.

The proxy functionality is used for querying Prometheus directly:

```
curl
'http://198.51.100.125:19091/api/v1/rbfs/elements/rtbrick/services/prometheus/prox
y/api/v1/query?query=up' | jq .
```

But it is also used for federation and therefore the following URL is used:

```
http://198.51.100.125:19091/api/v1/rbfs/elements/rtbrick/services/prometheus/proxy
/federate
```

Alert distribution

CTRLD can forward the alerts from the alertmanager to Graylog or any other REST endpoint.

API for Configuration

CTRLD provides a REST API Endpoint for the configuration of alerts and metrics.

Federation deployment model

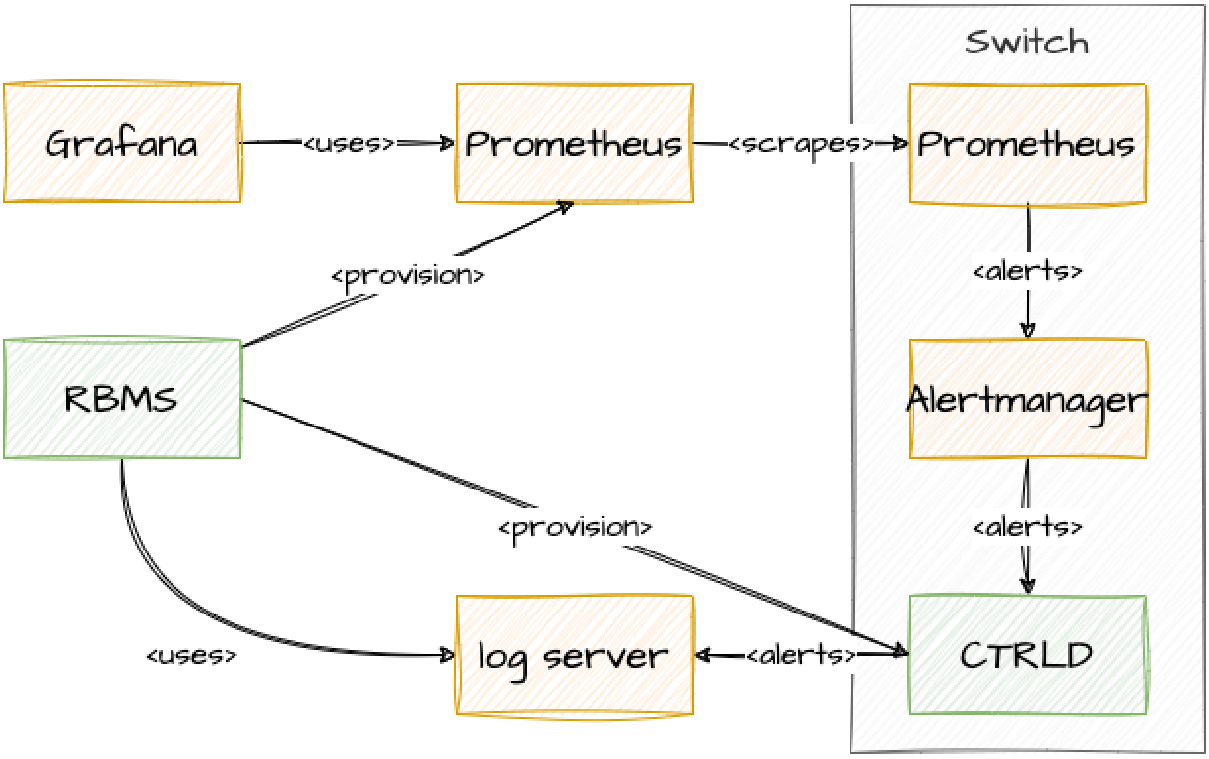


Figure 29. Federation of Prometheus, Alertmanager and graylog target

Prometheus is intended to have at least one instance per datacenter usually; also with a global Prometheus for global graphing or alerting. Federation allows for pulling metrics and aggregations up the hierarchy.

In the global Prometheus config, this time-series is pulled:

prometheus.yml:

```
global:
  scrape_interval: 60s # By default, scrape targets every 15 seconds.
  # A scrape configuration containing exactly one endpoint to scrape:
scrape_configs:
  - job_name: "federate"

    honor_labels: true
    metrics_path: '/federate'
    params:
      'match[]':
        - '{job="bds"}'
    scrape_interval: 15s
    # Patterns for files from which target groups are extracted.
    file_sd_configs:
      - files:
          - ./bds.target.yml
        refresh_interval: 5m
```

The match[] here requests all BDS job time series. By following this job naming convention, you do not have to adjust the config whenever there is a new aggregating rule.

The targets itself can be configured in a separate file.

bds.target.yml:

```
- targets: ['198.51.100.125:19091']
  labels:
    __metrics_path__:
"/api/v1/rbfs/elements/rtbrick/services/prometheus/proxy/federate"
    box: 125_rtbrick
```

16.1.2. TSDB Configuration

The following section describes how to configure the system to gather metrics and alerts out of the system.

Metric

To better understand the Data Model have a look at the Prometheus Data Model.

Metric Data Model

In RBFS it is possible to turn each table attribute into a metric.

> When you export the time-series metric data for an attribute which has more than 50 label values (user-defined, default labels), you may see truncated data in the exported metric.

The following table describes the configuration model:

Metric	
metric_name	Name of the metric (metric name conventions). That is the unique identifier for the metric.
table_name	Table Name for which the metric is designed, could also be a regular expression.
append_timestamp	Timestamp is epoch rendered in milliseconds and its value is equal to current metrics value’s creation time in RBFS.

bds_metric_type	• object-metric: if the metric should be gathered from regular table attributes • index-metric: if the metric should be gathered out of an attribute of an index table
index_name	Name of the index, if the bds_metric_type is index-metric.
metric_type	• gauge: is a metric that represents a single numerical value that can arbitrarily go up and down. Gauges are typically used for measured values like temperatures or current memory usage, but also "counts" that can go up and down, like the number of concurrent requests. • counter: is a cumulative metric that represents a single monotonically increasing counter whose value can only increase or be reset to zero on restart. For example, you can use a counter to represent the number of requests served, tasks completed, or errors. Do not use a counter to expose a value that can decrease. For example, do not use a counter for the number of currently running processes; instead use a gauge.
metric_description	Description of the metric.
attributes	List of Attributes (see Attribute Table) that will be streamed as metric.
filters	List of AttributeFilters (see AttributeFilter Table) that filters the table rows which should be considered for metric generation. Each filter in this list has to match in order to generate the metric, so the list implies an implicit AND.

Attribute	
attribute_name	Name of the attribute that should be streamed as metric. This Attribute has to be a numeric type, or a type that has a numeric converter.

filters	List of AttributeFilters (see the [tsdb:tsdb_config:::AttributeFilter] table) that filters the table rows which should be considered for metric generation. Each filter in this list has to match in order to generate the metric, so the list implies an implicit AND.
labels	List of AttributeLabels (see the [tsdb:tsdb_config:::AttributeLabel] table) that are attached to that metric.

AttributeFilter	
match_attribute_name	Attribute of the Table which is used to match against.
match_type	• exact: so the attribute has to match exactly the match value • regular-expression: the match value is a regular expression the attribute must match
match_value	The value that attribute has to match against.

AttributeLabel **CAUTION**: Remember that every unique combination of key-value label pairs represents a new time series, which can dramatically increase the amount of data stored. Do not use labels to store dimensions with high cardinality (many different label values), such as user IDs, email addresses, or other unbounded sets of values.	
label_name	Name of the Label (label name conventions).
dynamic	bool: If the label is dynamic, the label_value is treated as attribute_name, so the value of the attribute is used as the label value, otherwise the label value is used directly.
label_value	The value of the label or the attribute which should be used as label value.

filters	List of AttributeFilters (see [tsdb:tsdb_config:::AttributeFilter] Table) that filters the table rows which should be considered for label generation. Each filter in this list has to match in order to generate the label, so the list implies an implicit AND.

Configuring Metrics

The configuration of the Metrics can be done in various ways.

Configuring Metrics using Command Line Interface

To configure the Time Series Database, perform the following steps:

1. Define Metric configuration
2. Define Attribute configuration
3. Optional Filters at Metric Level and Attribute level
4. Defining labels to be attached to exported metric

Metric Configuration

Metric configuration is used to configure the parameters of the metric data being exported.

> Depending on the platform the exact resource name to be monitored can be found in global.chassis_0.resource.sensor, and adjust the Prometheus/Grafana configuration accordingly.

Syntax

```
set time-series metric <name>

set time-series metric <name> description <128 character description about the metric-name>

set time-series metric <name> prometheus-type <counter / gauge>

set time-series metric <name> bds-type <object-metric / index-metric>
set time-series metric <name> table-name <table-name>

set time-series metric <name> attribute <attribute-name>

set time-series metric <name> index-name <index-name>

set time-series metric <name> append-timestamp <true>

set time-series metric <name> filter <match-attribute-name>

set time-series metric <name> include-subscribed-tables [true / false]>
```

Command arguments

<metric-name>	Specifies the name of the metric exported, as would be reflected in Prometheus. Use the naming conventions as recommended by Prometheus
<128 character description about the metric-name >	Description of the metric
<counter / gauge>	Configures the metric data type. Currently the supported Prometheus metric data are: counter and gauge
<object-metric / index-metric >	Specifies the type of attribute, that is scraped and exported. There are two types, object-metric and index-metric
<table-name>	Specifies the target table, from which the data is scraped and exported.
<attribute-name>	Specifies the name of the attribute, in the target table to be scraped and exported
<append-timestamp>	Set the append-timestamp to true for exporting the metric values with timestamp. By default, the value is 'false'.
<index-name>	Specifies the index-name of the index-metric attribute. This configuration is applicable for index-metric alone.

<match-attribute-name>	Specifies the matching attribute name for the filter
include-subscribed-tables [true / false]	Specifies whether the configuration needs to be applied on a subscribed tables as well. Default: false.

Example

```
admin@rtbrick: cfg> set time-series metric chassis_fan_speed_rpm
admin@rtbrick: cfg> set time-series metric chassis_fan_speed_rpm table-name
global.chassis_0.resource.sensor
admin@rtbrick: cfg> set time-series metric chassis_fan_speed_rpm bds-type object-
metric
admin@rtbrick: cfg> set time-series metric chassis_fan_speed_rpm prometheus-type
gauge
admin@rtbrick: cfg> set time-series metric chassis_fan_speed_rpm description
"Chassis fan speed in rpm"
admin@rtbrick: cfg> set time-series metric chassis_fan_speed_rpm attribute rpm
admin@rtbrick: cfg> set time-series metric chassis_fan_speed_rpm include-
subscribed-table false
```

Allowed Attribute Types (Type Converters)

Normally only attributes are allowed, which are of type numeric, but for some types, there are built-in type converters, which allow also to use attributes of their types.

For the following BDS types, built-in type converters are provided by BDS. As per Prometheus data model, type converter will convert the BDS type into a 64bit float number.

BDS data type	Outcome number represents
unix-wallclock-timestamp	Seconds
unix-usec-wallclock-timestamp	Seconds
unix-usec-monotonic-timestamp	Seconds
unix-usec-coarse-wallclock-timestamp	Seconds
bandwidth	bps(bit per second)
temperature	Degree Celsius

Metric Filter Configuration

Metric filter configuration is used to configure the parameters of the filter. It is used to filter the exported metric. This is an optional configuration.

Syntax

set time-series metric <name> **filter** <match-attribute-name>

set time-series metric <name> **filter** <match-attribute-name> **match-type** <exact / regular-expression>

set time-series metric <name> **filter** <match-attribute-name> **match-attribute-value** <match-attribute-value>

Command arguments

<match-attribute-name>	Specifies the filter that filters the exported metric, based on specified criteria. This is optional configuration.
< exact / regular-expression >	Specifies the match type to be used, There are two options, exact and regular-expression.
<match-attribute-value>	Specifies the attribute value used for match. Fixed value for exact. Regex pattern for regular-expression

Example

Exact Value

```
admin@rtbrick: cfg> set time-series metric chassis_fan_speed_rpm filter
resource_type match-attribute-value fan
admin@rtbrick: cfg> set time-series metric chassis_fan_speed_rpm filter
resource_type match-type exact
```

Regular Expression

```
admin@rtbrick: cfg> set time-series metric chassis_fan_speed_rpm filter
resource_name match-attribute-value Chassis.*
admin@rtbrick: cfg> set time-series metric chassis_fan_speed_rpm filter
resource_name match-type regular-expression
```

Metric Attribute Label Configuration

Metric attribute config is used to configure the labels to be attached to the exported metric.

Syntax

set time-series metric <name> **attribute** <attribute-name> **label** <label-name>

set time-series metric <name> **attribute** <attribute-name> **label** <label-name> **label-type** <dynamic / static>

set time-series metric <name> **attribute** <attribute-name> **label** <label-name> **label-value** <label-value>

Command arguments

<label-name>	Specifies the name of label. User definable, Please use naming conventions as recommended by Prometheus
<dynamic / static>	Specifies the type of labels, a static value or dynamic value to be added.
<label-value>	Specifies the label-value to be used.

Example

Dynamic Label

```
admin@rtbrick: cfg> set time-series metric chassis_fan_speed_rpm attribute rpm
label fan
admin@rtbrick: cfg> set time-series metric chassis_fan_speed_rpm attribute rpm
label fan label-value resource_name
admin@rtbrick: cfg> set time-series metric chassis_fan_speed_rpm attribute rpm
```

```
label fan label-type dynamic
```

Static Label

```
admin@rtbrick: cfg> set time-series metric chassis_fan_speed_rpm attribute rpm
label vender
admin@rtbrick: cfg> set time-series metric chassis_fan_speed_rpm attribute rpm
label fan label-value rtbrick
admin@rtbrick: cfg> set time-series metric chassis_fan_speed_rpm attribute rpm
label fan label-type static
```

Metric Attribute Filter Configuration

Attribute filter config is used to configure the parameters of Attribute filter. It is used to filter the exported metric based on certain fields of the attribute. This is an optional configuration.

Syntax

set time-series metric <name> **attribute** <attribute-name> **filter** <match-attribute-name>

set time-series metric <name> **attribute** <attribute-name> **filter** <match-attribute-name> **match-type** <exact/regular-expression>

set time-series metric <name> **attribute** <attribute-name> **filter** <match-attribute-name> **match-value** <match-attribute-value>

Command arguments

<attribute name>	Specifies the filter that filters the exported metric , based on criteria of the attribute. This is optional config.
<exact / regular-expression>	Specifies the match type to be used, There are two options, exact and regular-expression.
<match-attribute-value>	Specifies the attribute value used for match. Fixed value for exact. Regex pattern for regular-expression

Example

The below example shows, the metric attribute will be exported only if the port_stat_if_in_discards is exactly 0.

```
admin@rtbrick: cfg> set time-series metric interface_statistics_data attribute
port_stat_if_in_ucast_pkts filter port_stat_if_in_discards
admin@rtbrick: cfg> set time-series metric interface_statistics_data attribute
port_stat_if_in_ucast_pkts filter port_stat_if_in_discards match-type exact
admin@rtbrick: cfg> set time-series metric interface_statistics_data attribute
port_stat_if_in_ucast_pkts filter port_stat_if_in_discards match-attribute-value 0
```

Metric Label Filter Configuration

Label filter configuration is used to set filter parameters that can be used to attach label based on certain criteria. This is an optional configuration.

Syntax

set time-series metric <name> **attribute** <attribute-name> **label** <label-key> **filter** <match-attribute-name>

set time-series metric <name> **attribute** <attribute-name> **label** <label-key> **filter** <match-attribute-name> **match-type** <regular-expression/exact>

set time-series metric <name> **attribute** <attribute-name> **label** <label-key> **filter** <match-attribute-name> **match-attribute-value** <match-attribute-value>

Command arguments

<match-attribute-name>	Specifies the filter that filters the exported metric, based on some attribute value.This is optional config.
< exact / regular-expression >	Specifies the match type to be used, There are two options, exact and regular-expression.
<match-attribute-value>	Specifies the attribute value used for match. Fixed value for exact. Regex pattern for regular-expression

Example

The below example sets label, interface_orientation to the exported data, only if the interface_name matches ifp-0/0/50.

```
admin@rtbrick: cfg> set time-series metric interface_statistics_data attribute
port_stat_if_in_ucast_pkts label interface_orientation
admin@rtbrick: cfg> set time-series metric interface_statistics_data attribute
port_stat_if_in_ucast_pkts label interface_orientation filter interface_name
admin@rtbrick: cfg> set time-series metric interface_statistics_data attribute
port_stat_if_in_ucast_pkts label interface_orientation filter interface_name
match-type exact
admin@rtbrick: cfg> set time-series metric interface_statistics_data attribute
port_stat_if_in_ucast_pkts label interface_orientation filter interface_name
match-attribute-value  ifp-0/0/50
```

Alert

RBFS uses the prometheus alerting feature to generate alerts. These alerts are forwarded to an alertmanager instance inside the rbfs container. The alertmanager instance sends the alert to CTRLD which distributes the alert to an HTTP Endpoint.

Alerts are also configured in a BDS table, and they are exported to Prometheus by the system.

Alert Data Model

Alert	
name	The name of the alert rule. That is the unique identifier for the rule.
group	Name of the alert group the alert belongs to. The alert group helps to structure the alerts.

interval	How often the rule should be evaluated. Pattern:"[0-9]+(ms |[smhdwy]" Example:"5s" In Prometheus the the interval can specified per alert group. So the alert alert group for Prometheus is calculated via {alert_group}_{interval}.
expr	Alert evaluation expression in promql
labels	Key, Value pairs of labels that should be applied. The labels clause allows specifying a set of additional labels to be attached to the alert. Any existing conflicting labels will be overwritten. The label values can be templated (see templating).
annotations	Key, Value pairs of annotations that should be applied. The annotations clause specifies a set of informational labels that can be used to store longer additional information such as alert descriptions or runbook links. The annotation values can be templated (see templating)
for	Alerts are considered firing once they have been returned for this long. Alerts which have not yet fired for long enough are considered pending. Pattern:"[0-9]+(ms |[smhdwy]" Example:"30s"
level	This is an explicit annotation label with the label name level. This is used to specify the severity: 1.Alert The annotation value can be templated (see templating)
summary	This is an explicit annotation label with the label name summary. The annotation values can be templated (see templating).

description	This is an explicit annotation label with the label name description. The annotation values can be templated (see templating).

Configuration

The configuration of the Metrics can be done in various ways.

Configuring Alert Using CLI

Syntax

```
set time-series alert <name>

set time-series alert <name> group <group>

set time-series alert <name> for <for>

set time-series alert <name> interval <interval>

set time-series alert <name> expr <expr>

set time-series alert <name> level <level>

set time-series alert <name> summary <summary>

set time-series alert <name> description <description>

set time-series alert <name> labels <label>

set time-series alert <name> annotations <annotations>
```

Command arguments

<name>	The name of the alert rule. That is the unique identifier for the rule.
<group>	Name of the alert group the alert belongs to. The alert group helps to structure the alerts.

<name>	**The name of the alert rule. That is the unique identifier for the rule.**
<interval>	How often the rule should be evaluated. Pattern:"[0-9]+(ms |[smhdwy]" Example:"5s" In Prometheus the the interval can specified per alert group. So the alert alert group for Prometheus is calculated via {alert_group}_{interval}.
<expr>	Alert evaluation expression in promql
<label>	Key, Value pairs of labels that should be applied. The labels clause allows specifying a set of additional labels to be attached to the alert. Any existing conflicting labels will be overwritten. The label values can be templated (see templating).
<annotations>	Key, Value pairs of annotations that should be applied. The annotations clause specifies a set of informational labels that can be used to store longer additional information such as alert descriptions or runbook links. The annotation values can be templated (see templating)
<for>	Alerts are considered firing once they have been returned for this long. Alerts which have not yet fired for long enough are considered pending. Pattern:"[0-9]+(ms |[smhdwy]" Example:"30s"
<level>	This is an explicit annotation label with the label name level. This is used to specify the severity: 1.Alert The annotation value can be templated (see templating)

<name>	The name of the alert rule. That is the unique identifier for the rule.
<summary>	This is an explicit annotation label with the label name summary. The annotation values can be templated (see templating).
<description>	This is an explicit annotation label with the label name description. The annotation values can be templated (see templating).

Example

```
admin@rtbrick: cfg> set time-series alert sample_alert
admin@rtbrick: cfg> set time-series alert sample_alert group hardware_metrics
admin@rtbrick: cfg> set time-series alert sample_alert for 30s
admin@rtbrick: cfg> set time-series alert sample_alert interval 5s
admin@rtbrick: cfg> set time-series alert sample_alert expr
avg_over_time(cpu_temperature_celcius[1m])>100
admin@rtbrick: cfg> set time-series alert sample_alert level 2
admin@rtbrick: cfg> set time-series alert sample_alert summary "Element {{
$labels.element_name }} CPU {{$labels.cpu}} HIGH temperature"
admin@rtbrick: cfg> set time-series alert sample_alert description "Cpu {{
$labels.cpu }} of element {{ $labels.element_name }} has a temperature o
ver 100 for more than 30 seconds"
admin@rtbrick: cfg> set time-series alert sample_alert labels device:leaf1
admin@rtbrick: cfg> set time-series alert sample_alert annotations "sample-
annotation-key:sample-value"
```

Enabling/Disabling Time Series Database History

In every Brick Daemon, the history of time series databases can be enabled. By default, time series database history is disabled.

Syntax

set time-series history-status <option>

Attribute	Description
[disable|enable]	Enable or disable time series database history. Time series database history is disabled, by default.

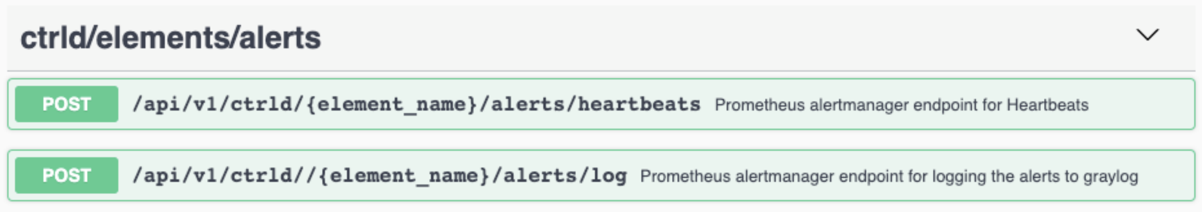
Example:

```
supervisor@S2-STD-7-7006>bm06-tst.fsn.rtbrick.net: cfg> show datastore confd table
global.time-series.config
```

```
Object: 0, Sequence 3, Last update: Mon May 23 09:02:28 GMT +0000 2022
  Attribute                                 Type                           Length
Value
  configuration_name (1)                    string (9)                          8
rtbrick
  time-series-history-enable (2)            boolean (6)                         1
False
```

Graylog Alert Distribution

The alertmanager on RBFS is configured to send alerts to CTRLD.



ctrld/elements/alerts

POST `/api/v1/ctrld/{element_name}/alerts/heartbeats` Prometheus alertmanager endpoint for Heartbeats

POST `/api/v1/ctrld//{element_name}/alerts/log` Prometheus alertmanager endpoint for logging the alerts to graylog

CTRLD therefore has an endpoint where the alerts are sent to. CTRLD translates the notification and forwards the message to the configured log management system. The instance used for forwarding is "prometheus".

16.1.3. Installation

The RtBrick fullstack comes with a ready to use tsdb instance. So no more installation on RBFS has to be done.

For federation of metrics, a global prometheus instance is needed. To visualize the metrics a Grafana instance has to be installed, and to get the alert messages, a graylog instance has to be set up. This document does not contain an installation guide for that systems.

The information about configuring a federation Pprometheus to scrape metrics from a RBFS installation is described in the Federation deployment model section.

16.2. Resmon

16.2.1. Resource Monitoring (resmon) Overview

Monitoring the system resources is very crucial to analyze the health of devices. RBFS has a dedicated daemon called resmond to discover and monitor the device resources. Resmond polls the system resources to gather the status of the resource and store this data in the resource-specific BDS table.

The Resource Monitoring (resmon) functionality of RBFS provides support for monitoring the following components:

- CPU
- Memory
- Processes
- Disks
- Sensor
- Optics

CPU

Resmond collects CPU hardware information from the global.chassis_0.resource.cpu table. In addition, Resmond calculates CPU usage dynamically and stores this information in the global.chassis_0.resource.cpu_usage table.

Memory

Resmond collects RAM hardware information from the global.chassis_0.resource.mem table. In addition, Resmond gathers memory usage information in the global.chassis_0.resource.mem_usage table.

Processes

Resmond collects process usage information of Brick Daemon(BD) that runs in the RBFS and stores the information in the global.chassis_0.resource.proc_usage table. It dynamically gathers the process information and calculates the CPU and the memory usage of the individual Brick Daemons.

Disks

Resmond collects the disk information from the global.chassis_0.resource.disk table. In addition, Resmond collects disk usage information in the global.chassis_0.resource.disk_usage table.

Sensor

Resmond collects the reading data of the hardware sensor such as temperature, fan, power-supply, and system LED. The data collected from the sensor are stored in the global.chassis_0.resource.sensor table.

> The RBFS implementation supports pluggable optics modules on white box switches only.

System Clock

Resmond provides support for monitoring the system clock so that the system clock is always in sync with the NTP server clock. This ensures that the deviation from the NTP server clock always remains within acceptable limits. Resmond collects the system clock information from the global.os.timex table.

> The RBFS implementation supports pluggable optics modules on white box switches only.

Optical Modules

Resmond monitors optical transceivers plugged onto the chassis. It reads transceivers EEPROM (Electrically Erasable Programmable Read-Only Memory) data and translates the data to respective fields in the BDS tables.

Resmond provides the following functionalities for monitoring optical transceivers:

- Provides a mechanism to discover and monitor optics modules. Supported optics modules include SFP, SFP+, QSFP, QSFP+, and QSFP28 (DAC is not supported).
- Provides CLIs to write to optics modules
- Provides show commands to see optics inventory and status of each module
- Logs the status of the optics module

> The RBFS implementation supports monitoring of pluggable optics modules on white box switches only.

The following are some of the important tasks (but not limited to) that the Resmond application performs:

- Optics inventory: Identifying the following brief information of a discovered optics module and stores in table global.chassis_0.resource.optics.inventory.

    Port

    Type

    Vendor

    Serial Number

    Part Number

- Read the following optics data from a module and stores in the table: global.chassis_0.resource.optics.module.

    RX/TX alarming (loss of light and loss of signal)

- RX/TX power status

    Voltage and BIAS status

    Temperature

- Write the optics data to an optics module

    Enabling high power class on QSFP28

    Shutdown lasers (QSFP28, SFP+ and SFP)

Optics Logging

The Resmond can log the following Optics module events:

- Temperature high alarm
- Temperature high warning
- Temperature low alarm
- Temperature low warning
- Voltage high alarm
- Voltage high warning
- Voltage low alarm
- Voltage low warning
- Lane power high alarm
- Lane power high warning

- Lane power low alarm
- Lane power low warning
- Lane bias high alarm
- Lane bias high warning
- Lane bias low alarm
- Lane bias low warning

Q2C Resource Monitoring

Q2C platform resource-specific usage metrics are stored in the BDS table: local.bcm.q2c.resource.monitor. Resource usage information enables you to understand the scale of services that the device performs and how it optimally leverages the resource usage.

The following table provides the list of supported resource types for monitoring in RBFS.

Resource Type	Description
EEDB_L2TP	EEDB is an Egress Encapsulation Data Base. This resource is consumed when L2TP subscribers are created in hardware.
EEDB_MPLS_TUNNEL	EEDB is an Egress Encapsulation Data Base. This resource is consumed when MPLS tunnels are created in the chip.
EEDB_PPPOE	EEDB_PPPOE is used for PPPoE encapsulation. This resource is consumed when PPPoE subscribers are created in hardware.
EEDB_PWE	This resource is consumed when L2X or cross-connection sessions are created in hardware.
IN_AC_C_C_VLAN_DB	This resource is consumed when an ingress logical interface for double-tagged VLAN is created.
IN_AC_C_VLAN_DB	This resource is consumed when an ingress logical interface for single tag VLAN is created.
IN_AC_UNTAGGED_DB	This resource is consumed when an ingress logical interface for untagged IFLs is created.

Resource Type	Description
IPV4_MULTICAST_PRIVATE_LPM_FORWARD	LPM stands for Longest Prefix Match. This resource is consumed for multicast (source, group) entries.
IPV4_UNICAST_PRIVATE_LPM_FORWARD	This resource is consumed for non-default VRF instance IPv4 prefixes.
IPV4_UNICAST_PRIVATE_LPM_FORWARD_2	This resource is consumed for default VRF instance IPv4 prefixes.
IPV6_UNICAST_PRIVATE_LPM_FORWARD	This resource is consumed for non-default VRF instance IPv6 prefixes.
IPV6_UNICAST_PRIVATE_LPM_FORWARD_2	This resource is consumed for default VRF instance IPv6 prefixes.
L3_RIF	This resource is consumed for the L3 interfaces.
L2TPV2_DATA_MESSAGE_TT	This resource is consumed when the L2TP subscribers are created in hardware.
MPLS_FWD	This resource is consumed for MPLS entries for which forwarding actions are involved.
MPLS_TERMINATION_SINGLE_LABEL_DB	This resource is consumed for MPLS entries for which label termination is required.
MULTICAST_MCDB	This resource is consumed for multicast groups created in hardware.
PPPOE_O_ETH_TUNNEL_FULL_SA	This resource is consumed for PPPoE subscribers in hardware.

Example: Logical table information for the resource type EEDB_L2TP

```
supervisor@ufi08.q2c.u23.r4.nbg.rtbrick.net: dbg> bcm "dbal table info
table=EEDB_L2TP"

Logical table info  EEDB_L2TP
=============================

        Access method: MDB
        Table type: DIRECT
        Touched status: Initialized
        Entries Status: Max Capacity: HW dependent (see mapping), Committed 0
        Bulk mode range NOT supported
        Maturity_level: HIGH
        Table Labels: L2, L3, MPLS, EEDB
        Core mode: SBC
        Max key value: 1048575
        Max payload size in bits: 101
```

```
<...>
```

Example: Logical table information for the resource type EEDB_MPLS_TUNNELEEDB_MPLS_TUNNEL

```
supervisor@rtbrick>ufi07.q2c.u21.r4.nbg.rtbrick.net: dbg> bcm "dbal table info
table=EEDB_MPLS_TUNNEL"

Logical table info  EEDB_MPLS_TUNNEL
====================================

        Access method: MDB
        Table type: DIRECT
        Touched status: Initialized
        Entries Status: Max Capacity: HW dependent (see mapping), Committed 18
        Bulk mode range NOT supported
        Maturity_level: HIGH
        Table Labels: L2, L3, MPLS, EEDB
        Core mode: SBC
        Max key value: 1048575
        Max payload size in bits: 147
<...>
```

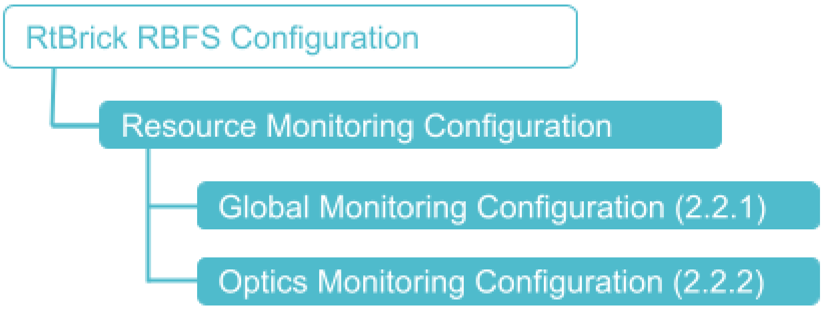
Supported Platforms

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

16.2.2. Resmon Configuration

Configuration Hierarchy

The diagram illustrates the Resmon configuration hierarchy.


RtBrick RBFS Configuration
Resource Monitoring Configuration
Global Monitoring Configuration (2.2.1)
Optics Monitoring Configuration (2.2.2)

Configuration Syntax and Commands

The following sections describe the Resmon configuration syntax and commands.

Global Monitoring Configuration

This command sets the poll-interval for the resmond to discover optics.

Syntax:

set resmon monitoring <poll-interval>

Attribute	Description
poll_interval <poll-interval>	Specifies the interval (in seconds) at which optics should be polled. The interval can range from 3 to 10000 seconds. The default interval is 10 seconds.

Optics Configuration

You can use this command to disable or enable (By, default enabled) tx laser or high-power class of an optics module on a specific interface.

Syntax:

set resmon optics <interface> …

Attribute	Description
interface <interface-name>	Name of the interface
high-power-class [disable / enable]	Enable or disable high power class for optics module. Enabled, by default.
tx [disable / enable]	Enable or disable lasers for optics module. Enabled, by default.

16.2.3. Resmon Operational Commands

Resmon Show Commands

The Resmon show commands provide detailed information about the resources and their usage.

CPU Information

This command displays the CPU detail and CPU information.

Syntax:

show cpu <option>

Option	Description
summary	Displays the CPU information.
usage	Displays the CPU usage information

Example 1: CPU summary

```
supervisor@rtbrick: op> show cpu summary
CPU_0
  Vendor               : GenuineIntel
  Model                : Intel(R) Xeon(R) CPU D-1518 @ 2.20GHz
  Architecture         : x86_64
  Serial No            : 63 06 05 00 FF FB EB BF
  Clock(MHz)           : 1996.620
  BogoMIPS             : 4400.00
  Physical cores       : 4
  Logical cores        : 8
  Endian               : True
  Cache alignment      : 64 Bytes
  L1 data cache        : 32768 Bytes
  L1 instruction cache : 32768 Bytes
  L2 unified cache     : 262144 Bytes
  L3 unified cache     : 6291456 Bytes
  L4 unified cache     : 0 Bytes
supervisor@rtbrick: op>
```

Example 2: CPU usage information.

```
supervisor@rtbrick: op> show cpu usage
Name         Total       User     System       Nice   I/O Wait       Idle        IRQ   Soft IRQ
cpu            31%        23%         8%         0%         0%        68%         0%         0%
cpu0           11%         4%         5%         0%         0%        88%         0%         2%
cpu1          100%        63%        36%         0%         0%         0%         0%         0%
cpu2           54%        51%         2%         0%         0%        45%         0%         0%
cpu3           57%        55%         2%         0%         0%        42%         0%         0%
cpu4            6%         1%         5%         0%         0%        93%         0%         0%
cpu5            4%         2%         2%         0%         0%        95%         0%         0%
cpu6            8%         3%         4%         0%         0%        91%         0%         0%
cpu7            6%         1%         5%         0%         0%        94%         0%         0%
supervisor@rtbrick: op>
```

Memory Details

This command displays the memory details and usage information.

Syntax:

show memory <option>

Option	Description
summary	Displays the system memory information.
usage	Displays the memory usage information.

Example 1: System memory information

```
supervisor@rtbrick: op> show memory summary
System Memory
  Maximum capacity                 : 128 GB
  Error correction type            : Multi-bit ECC
  Number of memory slots available : 4
  Number of memory slots occupied  : 2
Bank              Size        Location          Type      Speed       Configured Speed   Vendor      Serial
No      Part No
NODE 1            8192 MB     DIMM_A1           DDR4      2133 MT/s   2133 MT/s          Undefined   00000002
TS1GSH72V1H
NODE 1            8192 MB     DIMM_B1           DDR4      2133 MT/s   2133 MT/s          Undefined   00000027
TS1GSH72V1H
```

Example 2: System memory usage information.

```
supervisor@rtbrick: op> show memory usage
Name     Total            Used             Free         Shared       Buffers      Cached
RAM      16.69 GB         4.54 GB          10.08 GB     578.17 MB    103.12 MB    2.06 GB
SWAP     0 bytes          0 bytes          0 bytes      n/a          n/a          n/a
```

Process Details

This command displays the process usage information.

Syntax:

show process usage <option>

Option	Description
process-id <pid>	Displays the process usage for the specified process identifier.

Option	Description
process-name <process-name>	Displays the process usage for the specified process.
summary	Displays the process usage summary information.

Example 1: Process usage summary information.

```
supervisor@rtbrick: op> show process usage summary
Name             PID         VIRT              Resident Memory       Sharable Memory     CPU Percentage
Memory Percentage   CPU Affinity
bgp.appd.1       4456        384.67 MB         122.69 MB             29.32 MB            0.81
0.74                0-7
bgp.iod.1        4469        694.63 MB         148.39 MB             30.22 MB            2.01
0.89                0-7
confd            213         1.24 GB           769.9 MB              31.89 MB            0.81
4.61                0-7
etcd             110         600.9 MB          196.4 MB              29.26 MB            1.21
1.18                0-7
fibd             288         13.68 GB          1.74 GB               210.79 MB           161.17
10.45               1
ifmd             147         496.74 MB         154.25 MB             29.99 MB            1.01
0.92                0-7
igmp.appd.1      4482        356.38 MB         100.29 MB             29.27 MB            1.01             0.6
0-7
igmp.iod.1       4495        540.14 MB         132.68 MB             29.41 MB            2.01             0.8
0-7
ipoed.1          175         503.51 MB         112.9 MB              29.3 MB             2.01
0.68                0-7
l2tpd.1          4508        475.71 MB         103.45 MB             29.26 MB            2.01
0.62                0-7
lldpd            128         368.32 MB         106.75 MB             29.29 MB            2.21
0.64                0-7
mribd            158         381.8 MB          115.57 MB             29.38 MB            1.01
0.69                0-7
```

Example 2: Process usage for the specified process.

```
supervisor@rtbrick: op> show process usage process-name fibd
Process Name: fibd
  PID                           : 288
  REST port                     : 5522
  Debug port                    : 5521
  Allowed CPU list              : 1
  CPU usage at user space       : 25964
  CPU usage at kernel space     : 12633
  CPU usage percentage          : 152.815678
  Memory usage percentage       : 10.452729
  Peak virtual memory usage     : 13355692
  Current virtual memory usage : 13.68 GB
  <...>
```

Example 3: Process usage for the specified process ID.

```
supervisor@rtbrick: op> show process usage process-id 4456
Process Name: bgp.appd.1
  PID                           : 4456
```

```
   REST port                      : 4102
   Debug port                     : 4101
   Allowed CPU list               : 0-7
   CPU usage at user space        : 20103
   CPU usage at kernel space      : 18011
   CPU usage percentage           : 1.011122
   Memory usage percentage        : 0.735218
   Peak virtual memory usage      : 375652
   Current virtual memory usage : 384.67 MB
   Locked virtual memory          : n/a
   RSS virtual memory             : 122.69 MB
   <...>
```

Sensor Details

This command displays the fan, power-supply, system-led, and temperature information.

Syntax:

show sensor <option>

Option	Description
fan	Displays information about the sensor fan.
power-supply	Displays the sensor power supply information.
system-led	Displays system LED information
temperature	Displays the sensor temperature information.
detail	You can specify detail at the end of any of the options above to display information in detail.

Example 1: Sensor temperature information.

```
supervisor@rtbrick: op> show sensor temperature
Name                     Temperature         Status
CPU Core                 54 °C               PRESENT
LM75-1-48                35 °C               PRESENT
LM75-2-49                30 °C               PRESENT
LM75-3-4A                33 °C               PRESENT
LM75-3-4B                30 °C               PRESENT
PSU-1 Thermal Sensor 1   31 °C               PRESENT
supervisor@rtbrick: op>
```

Example 2: Detailed information about the sensor temperature.

```
supervisor@rtbrick: op> show sensor temperature
```

```
Name                      Temperature         Status
CPU Core                  54 °C               PRESENT
LM75-1-48                 36 °C               PRESENT
LM75-2-49                 30 °C               PRESENT
LM75-3-4A                 33 °C               PRESENT
LM75-3-4B                 30 °C               PRESENT
PSU-1 Thermal Sensor 1    31 °C               PRESENT
supervisor@rtbrick: op>
```

Example 3: Information about sensor fan

```
supervisor@rtbrick: op> show sensor fan
Name                Fan Speed (rpm)     Status
PSU 1 - Fan 1       26496               PRESENT, F2B
Chassis Fan - 1     8700                PRESENT, F2B
Chassis Fan - 2     8700                PRESENT, F2B
Chassis Fan - 3     8700                PRESENT, F2B
Chassis Fan - 4     8700                PRESENT, F2B
Chassis Fan - 5     8700                PRESENT, F2B
Chassis Fan - 6     8600                PRESENT, F2B
supervisor@rtbrick: op>
```

Example 4: Detailed information about sensor fan.

```
supervisor@rtbrick: op> show sensor fan detail

PSU 1 - Fan 1
  Sensor resource ID   : 8388614
  Vendor               : n/a
  Model                : NULL
  Serial No            : n/a
  Status               : PRESENT, F2B
  Status code          : 9
  Fan speed            : 26496 rpm
  Location             : PSU 1
Chassis Fan - 1
  Sensor resource ID   : 8388608
  Vendor               : ALTERA
  Model                : 5M1270ZF256C5N
  Serial No            : n/a
  Status               : PRESENT, F2B
  Status code          : 9
  Fan speed            : 8700 rpm
  Location             : Fan Board
Chassis Fan - 2
  Sensor resource ID   : 8388609
  Vendor               : ALTERA
  Model                : 5M1270ZF256C5N
  Serial No            : n/a
  Status               : PRESENT, F2B
  Status code          : 9
  Fan speed            : 8800 rpm
  Location             : Fan Board
```

Example 5: Sensor power supply information

```
supervisor@rtbrick: op> show sensor power-supply
Name              Current In  Current Out Voltage In  Voltage Out Power In    Power Out   Status
PSU-1             0 mA        12109 mA    0 mV        11984 mV    0 mW        146000 mW   PRESENT
PSU-2             0 mA        0 mA        0 mV        0 mV        0 mW        0 mW        PRESENT, UNPLUGGED
supervisor@rtbrick: op>
```

Example 6: Detailed information about the sensor power supply.

```
supervisor@rtbrick: op> show sensor power-supply detail

PSU-1
  Sensor resource ID   : 16777216
  Vendor               : n/a
  Model                : YM-2651Y
  Serial No            : n/a
  Status               : PRESENT
  Status code          : 1
  Input current        : 0 mA
  Output current       : 12031 mA
  Input voltage        : 0 mV
  Output voltage       : 11984 mV
  Input power          : 0 mW
  Output power         : 144000 mW
  Location             : n/a
PSU-2
  Sensor resource ID   : 16777217
  Vendor               : n/a
  Model                : NULL
  Serial No            : n/a
  Status               : PRESENT, UNPLUGGED
  Status code          : 5
  Input current        : 0 mA
  Output current       : 0 mA
  Input voltage        : 0 mV
  Output voltage       : 0 mV
  Input power          : 0 mW
  Output power         : 0 mW
  Location             : n/a
supervisor@rtbrick: op>
```

Example 7: Sensor system LED information

```
supervisor@rtbrick: op> show sensor system-led
Name                     LED Mode         Status
Chassis LED 1 (LOC LED)  OFF              PRESENT
Chassis LED 5 (FAN LED)  AUTO             PRESENT, ON
Chassis LED 2 (DIAG LED) OFF              PRESENT
Chassis LED 3 (PSU1 LED) AUTO             PRESENT, ON
Chassis LED 4 (PSU2 LED) AUTO             PRESENT, ON
supervisor@rtbrick: op>
```

Example 8: Detailed information about the system LED

```
supervisor@rtbrick: op> show sensor system-led detail
```

```
Chassis LED 1 (LOC LED)
  Sensor resource ID   : 12582912
  LED mode             : OFF
  Status               : PRESENT
  Status code          : 1
  Capability           : ON_OFF, ORANGE
  Capability code      : 4097
Chassis LED 5 (FAN LED)
  Sensor resource ID   : 12582916
  LED mode             : AUTO
  Status               : PRESENT, ON
  Status code          : 5
  Capability           : ON_OFF, AUTO
  Capability code      : 4194305
Chassis LED 2 (DIAG LED)
  Sensor resource ID   : 12582913
  LED mode             : OFF
  Status               : PRESENT
  Status code          : 1
  Capability           : ON_OFF, ORANGE, GREEN
  Capability code      : 69633
```

Optics Details

This command displays optics information for inventory and interface.

Syntax:

show optics <option>

Option	Description
interface <interface-name>	Displays optics information for the specified interface.
inventory	Displays optics inventory information.

Example: Optics information for the specified interface.

```
supervisor@rtbrick: op> show optics interface ifp-0/1/6
Physical Interface: ifp-0/1/6
Type                   : QSFP28
Description            : 100G-CWDM4
Connector Type         : Lucent Connector
Vendor                 : FS
Serial Number          : F2030882972
Part Number            : QSFP28-IR4-100G
Vendor Material Number : CMUIAMACAB10-3146-02
Power Class            : Class 4 (3.5W)
Power Class State      : HIGH
Wavelength             : 1310.000000
Lane Id                : 1
```

```
  Laser bias current                              : 30.824 mA
  Laser tx power                                  : 1.496 mW / 1.749 dbm
  Laser rx power                                  : 1.084 mW / 0.35 dbm
  Module temperature                              : 39.164 °C
  Module voltage                                  : 3.292 V
  Tx disable                                      : False
  High power class enable                         : True
  Laser Tx loss of signal                         : False
  Laser Tx loss of lock                           : False
  Laser Rx loss of signal                         : False
  Laser Rx loss of lock                           : False
  Laser bias current high alarm                   : False
  Laser bias current high warning                 : False
  Laser bias current low alarm                    : False
  Laser bias current low warning                  : False
  Module voltage high alarm                       : False
  Module voltage high warning                     : False
  <...>
```

Example 2: Optics inventory information.

```
supervisor@rtbrick: op> show optics inventory
Interface      Type     Description    Connector Type           Vendor         Part Number         Serial
Number   Material Number   Power Class      Power State
ifp-0/1/2      QSFP28   100GBASE-CR4   No Seperable connector   FS             Q28-PC005
G2110248549-1   n/a               Class 1 (1.5W)   LOW
ifp-0/1/3      QSFP28   100G-CWDM4     Lucent Connector         LambdaGain     LL1C31A2A           L12AA60027
T-UNIQSF40907038  Class 4 (3.5W)   HIGH
ifp-0/1/4      QSFP28   100GBASE-CR4   No Seperable connector   FS             Q28-PC01
G2006503747-2   n/a               Class 1 (1.5W)   LOW
ifp-0/1/5      QSFP28   100GBASE-LR4   Lucent Connector         LambdaGain     LL1S31B0A           L82A9S0018
T-UNIQSF40907039  Class 4 (3.5W)   HIGH
ifp-0/1/6      QSFP28   100G-CWDM4     Lucent Connector         FS             QSFP28-IR4-100G
F2030882972     CMUIAMACAB10-314  Class 4 (3.5W)   HIGH
ifp-0/1/8      QSFP28   100GBASE-CR4   No Seperable connector   FS             Q28-PC005
G2110248550-1   n/a               Class 1 (1.5W)   LOW
ifp-0/1/9      QSFP28   100GBASE-CR4   No Seperable connector   FS             Q28-PC005
G2110248551-1   n/a               Class 1 (1.5W)   LOW
ifp-0/1/11     QSFP28   100GBASE-CR4   No Seperable connector   Fiberstore     QSFP28-100G-DAC
F1800032252-2   n/a               Class 1 (1.5W)   LOW
ifp-0/1/14     QSFP28   100GBASE-CR4   No Seperable connector   FS             Q28-PC005
G2110248569-1   n/a               Class 1 (1.5W)   LOW
ifp-0/1/15     QSFP28   100GBASE-CR4   No Seperable connector   FS             Q28-PC01
G2006503745-1   n/a               Class 1 (1.5W)   LOW
ifp-0/1/18     QSFP28   100GBASE-CR4   No Seperable connector   FS             Q28-PC01
G2006503743-1   n/a               Class 1 (1.5W)   LOW
ifp-0/1/20     QSFP28   100G-CWDM4     Lucent Connector         LambdaGain     LL1C31A2A           L12AB20043
T-UNIQSF40907038  Class 4 (3.5W)   HIGH
ifp-0/1/27     QSFP28   100GBASE-LR4   Lucent Connector         LambdaGain     LL1S31B0A           L82A9S0025
T-UNIQSF40907039  Class 4 (3.5W)   HIGH
ifp-0/1/28     QSFP28   100GBASE-CR4   No Seperable connector   FS             Q28-PC01
G1810038309-1   n/a               Class 1 (1.5W)   LOW
ifp-0/1/31     QSFP28   100GBASE-CR4   No Seperable connector   FS             Q28-PC01
G1810038285-2   n/a               Class 1 (1.5W)   LOW
ifp-0/0/0      SFP      UNKNOWN        Copper Pigtail           FS             SFPP-PC015
S2108004616-2   n/a               Class 1 (1.5W)   LOW
ifp-0/0/1      SFP      UNKNOWN        Copper Pigtail           FS             SFPP-PC015
G2006548415-2   n/a               Class 1 (1.5W)   LOW
```

Disk Details

This command displays disk information.

Syntax:

show disk <option>

Option	Description
summary	Displays the disk information.
usage	Displays the disk usage information.

Example: Summary of disks and their partitions.

```
supervisor@rtbrick: op> show disk summary
sda
  Size    : 29.8G
  Vendor  : ATA
  Model   : TS32ZBTMM1600
  Partitions
  Name               Size        Mountpoint
  sda1               256M        n/a
  sda2               128M        n/a
  sda3               2G          n/a
  sda4               128M        n/a
  sda5               128M        n/a
  sda6               2G          n/a
  sda7               25.2G       /platform
supervisor@rtbrick: op>
```

Example 2: Summary of disk usage information.

```
supervisor@rtbrick: op> show disk usage
Filesystem               Type             Size         Used         Available    Mountpoint          Usage %
none                     tmpfs            492 KB       0 bytes      492 KB       /dev                0.0
tmpfs                    tmpfs            8.15 GB      17.38 MB     8.13 GB      /run                0.21
tmpfs                    tmpfs            6.29 GB      483.6 MB     5.81 GB      /shm                7.69
tmpfs                    tmpfs            8.15 GB      62.74 MB     8.09 GB      /dev/shm            0.77
tmpfs                    tmpfs            5.12 MB      0 bytes      5.12 MB      /run/lock           0.0
devtmpfs                 devtmpfs         1.02 MB      0 bytes      1.02 MB      /dev/mem            0.0
/dev/sda7                ext4             25.87 GB     4.47 GB      20.06 GB     /var/log            18.21
tmpfs                    tmpfs            1.63 GB      0 bytes      1.63 GB      /run/user/1000      0.0
tmpfs                    tmpfs            8.15 GB      0 bytes      8.15 GB      /sys/fs/cgroup      0.0
tmpfs                    tmpfs            1.63 GB      696 KB       1.63 GB      /var/run-ext/onl/r  0.04
/var/cache/rtbrick/imag  overlay          25.87 GB     4.47 GB      20.06 GB     /                   18.21
supervisor@rtbrick: op>
```

Platform Details

This command displays platform information.

Syntax:

show platform

Example: Platform information

```
supervisor@rtbrick: op> show platform
x86_64-accton_as7316_26xb-r0
  Vendor                : Edgecore
  Manufacturer          : Accton
  Manufacture date      : 05/07/2021 16:55:51
  MAC address           : 90:3c:b3:16:00:00
  Part number           : F0PZZ5626002A-MACDDR-Nanya_NT5AD256M16D4_HRI
  Serial number         : AAB2115AACA
  Product name          : 7316-26XB-O-AC-F
  Onie version          : 2019.11.00.07
  Label revision        : R01C
  Diag version          : 01.01.00.03
  Country code          : TW
supervisor@rtbrick: op>
```

Hardware Resource Usage Information

The command show hardware resource monitor displays the hardware consumption information for the various resources.

Syntax:

show hardware resource monitor

Example: Hardware resource monitoring information

```
supervisor@rtbrick: op> show hardware resource monitor
   Resource Type                                          Consumed Estimated
   EEDB_L2TP                                                     0 49152
   EEDB_MPLS_TUNNEL                                              0 49152
   EEDB_PPPOE                                                    0 49152
   EEDB_PWE                                                      0 49152
   IN_AC_C_C_VLAN_DB                                             0 60292
   IN_AC_C_VLAN_DB                                              36 60292
   IN_AC_UNTAGGED_DB                                             3 60292
   IPV4_MULTICAST_PRIVATE_LPM_FORWARD                            0 568320
   IPV4_UNICAST_PRIVATE_LPM_FORWARD                             11 1136640
   IPV4_UNICAST_PRIVATE_LPM_FORWARD_2                           11 378880
   IPV6_UNICAST_PRIVATE_LPM_FORWARD                              7 568320
   IPV6_UNICAST_PRIVATE_LPM_FORWARD_2                           11 189440
   KBP_IPV4_UNICAST_PRIVATE_LPM_FORWARD                          0 0
   KBP_IPV6_UNICAST_PRIVATE_LPM_FORWARD                          0 0
   L2TPV2_DATA_MESSAGE_TT                                        0 60292
   L3_RIF                                                       51 40959
   MPLS_FWD                                                      0 1017112
   MPLS_TERMINATION_SINGLE_LABEL_DB                             14 60292
   MULTICAST_MCDB                                                2 262143
   PPPOE_O_ETH_TUNNEL_FULL_SA                                    0 24576
```

16.3. SNMP

16.3.1. SNMP Overview

SNMP (Simple Network Management Protocol) provides a network monitoring mechanism that collects state information from various network devices and components. The protocol enables you to monitor the RBFS network and detect network faults on remote devices.

SNMP can monitor interfaces, CPU usage, temperature of the device, bandwidth usage, and so on. For example, if an interface goes down on one of the devices, SNMP can quickly alert the change.

Understanding RBFS SNMP Implementation

The RBFS SNMP implementation allows retrieving system state information using the Protocol Data Unit (PDU) from various network components. SNMP defines system objects in Management Information Bases (MIBs), where each object forms a Protocol Data Unit (PDU).

A device that is SNMP enabled is known as SNMP agent. SNMP agent collects information from various devices and components and stores the data within MIB. An SNMP agent includes several objects such as interfaces and routing tables which can be interacted with. Every object has a unique identifier which is known as OID (Object Identifier). OIDs provide a unique identity for managed objects in an MIB hierarchy.

RBFS implements the SNMP daemon (snmpd) that maps the operational state API to SNMP MIBs for retrieving system state data.

SNMP Operations

SNMP allows performing various operations that include GET for retrieving data, SET for modifying data, TRAP for notifying an event and so on. These operations provide management access to the MIB hierarchy.

1. **GET**: The SNMP GET operation retrieves data from the managed device’s MIB.
2. **GETNEXT**: The GETNEXT operation retrieves data for the next object from the tree of objects on the device.
3. **SET**: The SNMP SET operation allows modifying data for a device.

4. **TRAP**: Event notification that is not requested.

> RBFS does not currently support the SNMP SET operation.

Supported SNMP MIBs

Management Information Base (MIB) is a collection of data that is organized hierarchically. You can define an MIB mapping for the supported MIBs. This MIB mapping provides a way to retrieve the required state information using the Operational State API or any other RBFS API or Prometheus.

Currently, RBFS supports two SNMP MIBs: the Interface MIB and the Host Resources MIBs.

Host resource MIB: Host resource MIB collects and provides information for host computer resources.

Interfaces MIB: Interfaces MIB retrieves information about the state of interfaces in devices.

Supported SNMP Versions

RBFS supports the SNMP version 2c and SNMP version 3. SNMP version 3 provides support for authentication and encryption and the data can be accessed after capture only by authorized users. You must choose either version 2c or version 3 before configuring other functionalities.

Information about SNMP v2c and v3

SNMP v2c allows access control through Communities, but the community information is not protected. Anyone with the community name can access all the information available for that community.

SNMP v3 provides a higher level of security using authentication and privacy protocols. RBFS supports the user-based security model defined in RFC 3414.

The user credentials verify the authenticity and integrity of the message by adding message authentication codes (MACs) to the SNMP packet. The privacy protocol allows for the encryption of the transmitted data.

16.3.2. SNMP Configuration

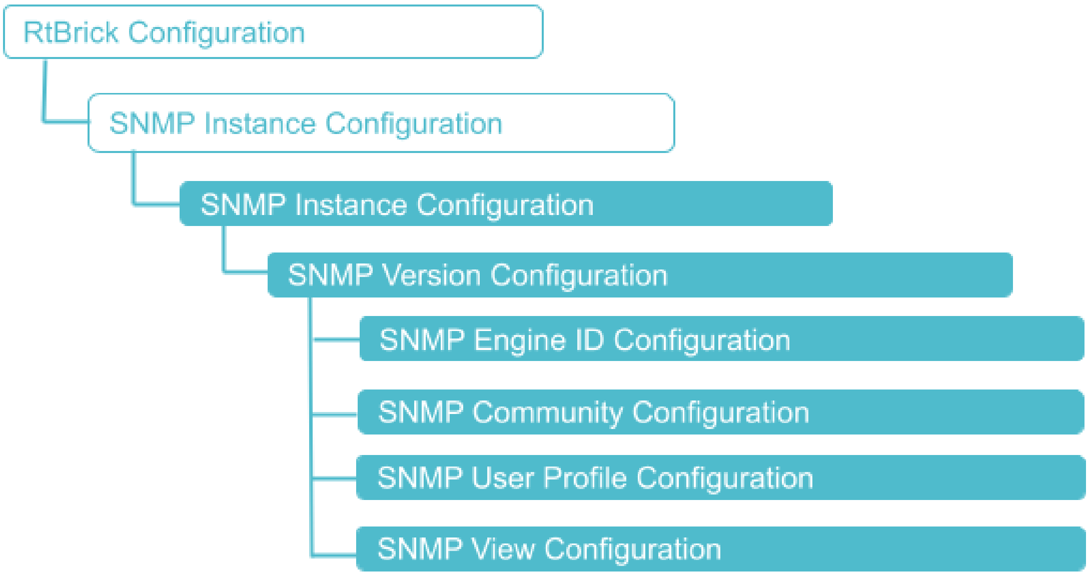
By default, SNMP is not enabled. To enable SNMP, you must complete the SNMP configurations.

RBFS supports SNMP 2c and 3 versions. You must first configure the desired version before configuring other functionalities. RBFS does not support running both the SNMP 2c and 3 versions at the same time.

The RBFS CLI displays all options regardless of the selected version. Even though functionalities such as Community can be configured for both of the versions, it works only in SNMP 2c version. You can define 'user profiles' on SNMP v3 and 'Community' in version 2c. Similarly, Engine ID can be defined in SNMP version 3.

Configuration Hierarchy

The diagram illustrates the SNMP configuration hierarchy. All SNMP configurations are performed within an instance.


RtBrick Configuration
SNMP Instance Configuration
SNMP Instance Configuration
SNMP Version Configuration
SNMP Engine ID Configuration
SNMP Community Configuration
SNMP User Profile Configuration
SNMP View Configuration

Configuration Syntax and Commands

The following sections describe the SNMP configuration syntax and commands.

SNMP Instance Configuration

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

Syntax

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

Attribute	Description
version	Specify the SNMP version. RBFS supports SNMP version 2c and version 3. You must first configure the desired version before configuring other functionalities.
engine-id	Specify the unique SNMP engine identifier. This is optional. If not specified, the system retrieves the default engine ID from the management port MAC address. Every SNMP v3 agent includes an engine ID that is a unique identifier for the agent. The engine ID is used to provide a higher level of security using authentication and encryption for SNMP v3 messages.

Example: SNMP Version and Engine Identifier Configuration

The following commands configure SNMP version 3 and engine ID: 268956.

```
set instance default protocol snmp version 3
set instance default protocol snmp engine-id 268956
```

The following example shows the SNMP version and engine ID configurations.

```
supervisor@rtbrick.net: cfg> show config instance default protocol snmp
{
  "rtbrick-config:snmp": {
    "version": "3",
    "engine-id": "268956"
  }
}
```

SNMP Community Configuration

An SNMP community can be defined only in the SNMP version 2c.

Syntax

set instance <instance-name> **protocol snmp community**

Attribute	Description
access-mode	Specify the access mode. Read, write and append are modes of access. 'ReadOnly' is the currently supported access mode.
view	Specify the list of view identifiers. View is optional. For information about Views, see section "2.2.3 SNMP View Configuration".

Example: SNMP v2c Community Configuration

The following commands configure a Community named 'public' with read-only access right to the 'interfaces' View.

```
set instance default protocol snmp version 2c
set instance default protocol snmp community public access-mode ReadOnly
set instance default protocol snmp community public view Interfaces
```

The following example shows SNMP v2c community configurations.

```
supervisor@rtbrick: cfg> show config instance default protocol snmp
{
  "rtbrick-config:snmp": {
    "version": "2c",
    "community": [
      {
        "name": "public"
        "access-mode": "ReadOnly"
      }
    ]
  }
}
```

SNMP View Configuration

An SNMP View is a subset of MIB objects. Views allow you to restrict access to certain items in the SNMP PDUs. You can restrict user and community access to certain attributes by defining views. A view restricts access to the PDUs included in the View. If the access is not restricted by views, the user or community is allowed to view all data available through SNMP.

Syntax

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

Attribute	Description
include <include>	List of OID patterns that are included in the view.
instance	List of instances. It restricts the view to the specified instances. If no instance is defined, the view can access to all instances.

Example: SNMP View Configuration

The following commands configure SNMP View. In this example configuration, SNMP version has been specified as 2c and 'View' name is specified as interfaces. The 'interfaces' view includes the OID 1.3.6.1.2.1.2.* in the view list. In addition, the configuration shows a user 'community' named public has been configured and the community has read-only access to the View.

```
set instance default protocol snmp version 2c
set instance default protocol snmp view interfaces include 1.3.6.1.2.1.2.*
set instance default protocol snmp community public access-mode ReadOnly
```

The following example shows the SNMP View configuration.

```
supervisor@rtbrick: cfg> show config instance default protocol snmp
{
  "rtbrick-config:snmp": {
    "version": "2c",
    "view": [
      {
        "name": "interfaces",
        "include": [
          "1.3.6.1.2.1.2.*"
          ]
      }
    ],
    "community": [
      {
        "name": "public",
        "access-mode": "ReadOnly",
        "view": [
          "interfaces"
          ]
        }
      ]
    }
  }
```

SNMP Trap Configuration

The trap is a notification about a specific condition or event that occurs on the device. RBFS supports event notifications for events: link up and link down. Unlike other PDU types, a trap is a notification that is sent without any request from the SNMP manager.

Syntax

set instance <instance-name> **protocol snmp trap** <trap-name> <attribute> <value>

Attribute	Description
community	Community is supported in SNMPv2c only. Specify the 'Community' to enable trap notification for SNMPv2c.
include	List of OIDs that are included for the trap notifications.
instance	Specify the instance.
trap-receiver	Specify trap receiver device.
user-profile	User profile is supported on SNMPv3 only. Specify the 'user profile' to enable trap notification for SNMPv3.

SNMP User Profile Configuration

You can create user profiles for SNMP version 3. It allows you to define login credentials, authentication methods, and privacy control.

Syntax

set instance <instance-name> **protocol snmp user-profile**

Attribute	Description
authentication-protocol	Specify SNMP authentication protocol. MD5, NoAuth, SHA, SHA224, SHA256, SHA384, and SHA512 are the supported authentication protocol.
password-encrypted-text	Specify SNMP user password in encrypted text.
password-plain-text	Specify SNMP user password in plain text.
privacy-password-encrypted-text	Specify SNMP privacy password in encrypted text.
privacy-password-plain-text	Specify SNMP privacy password in plain text.
privacy-protocol	Specify SNMP privacy protocol. Supported privacy protocols include AES192, AES192C, AES256, AES256C, DES, and NoPriv.
security-level	Specify SNMP v3 security level. Security levels exist only in SNMP v3. The following security levels are supported: • noAuthNoPriv: no authentication, no privacy • authNoPriv: authentication, no privacy • authPriv: authentication, privacy
view	Specify SNMP view list.

Example: SNMP User Profile Configuration

The following commands configure SNMP user profile. At first, SNMP Version 3 is configured with the user profile name as operator. Password type has been selected as password encrypted text. In this configuration, the security level is configured as AuthNoPriv and MD5 as type of the authentication protocol.

```
set instance default protocol snmp version 3
set instance default protocol snmp user-profile operator
set instance default protocol snmp user-profile operator password-encrypted-text
$2a6fd7db50a18a9f1f16b5c5b4214fab0
set instance default protocol snmp user-profile operator security-level AuthNoPriv
set instance default protocol snmp user-profile operator authentication-protocol
MD5
```

The following example shows the SNMP User Profile Configuration

```
supervisor@rtbrick: cfg> show config instance default protocol snmp
{
  "rtbrick-config:snmp": {
    "version": "3",
    "user-profile": [
      {
        "name": "operator",
        "password-encrypted-text": "$2a6fd7db50a18a9f1f16b5c5b4214fab0",
        "security-level": "AuthNoPriv",
        "authentication-protocol": "MD5"
      }
    ]
  }
}
```

Examples for SNMP Walk Operation

SNMP v2c SNMP Walk Output

The following is a sample output for the SNMP Walk for SNMP version 2c. SNMP version 2c has been configured with Community name as 'public' and, host IP address as 10.200.134.25.

```
snmpwalk -v 2c -c public 10.200.134.25
iso.3.6.1.2.1.2.2.1.1.1 = INTEGER: 1
iso.3.6.1.2.1.2.2.1.1.2 = INTEGER: 2
iso.3.6.1.2.1.2.2.1.1.3 = INTEGER: 3
iso.3.6.1.2.1.2.2.1.1.4 = INTEGER: 4
iso.3.6.1.2.1.2.2.1.1.5 = INTEGER: 5
iso.3.6.1.2.1.2.2.1.1.6 = INTEGER: 6
iso.3.6.1.2.1.2.2.1.1.7 = INTEGER: 7
iso.3.6.1.2.1.2.2.1.1.8 = INTEGER: 8
iso.3.6.1.2.1.2.2.1.1.9 = INTEGER: 9
iso.3.6.1.2.1.2.2.1.1.10 = INTEGER: 10
iso.3.6.1.2.1.2.2.1.1.11 = INTEGER: 11
iso.3.6.1.2.1.2.2.1.1.12 = INTEGER: 12
iso.3.6.1.2.1.2.2.1.1.13 = INTEGER: 13
iso.3.6.1.2.1.2.2.1.1.14 = INTEGER: 14
iso.3.6.1.2.1.2.2.1.1.15 = INTEGER: 15
iso.3.6.1.2.1.2.2.1.1.16 = INTEGER: 16
iso.3.6.1.2.1.2.2.1.1.17 = INTEGER: 17
iso.3.6.1.2.1.2.2.1.1.18 = INTEGER: 18
iso.3.6.1.2.1.2.2.1.1.19 = INTEGER: 19
iso.3.6.1.2.1.2.2.1.1.20 = INTEGER: 20
iso.3.6.1.2.1.2.2.1.1.21 = INTEGER: 21
iso.3.6.1.2.1.2.2.1.1.22 = INTEGER: 22
iso.3.6.1.2.1.2.2.1.1.23 = INTEGER: 23
<...>
```

SNMP v3 SNMP Walk Output

The following is a sample output for the SNMP Walk for SNMP version 3. SNMP version 3 has been configured with user as 'operator', MD5 as the authentication protocol, authNoPriv as the security level, and 10.200.134.25 as the host IP address.

```
snmpwalk -v 3 -u operator -A operator -a MD5 -l authNoPriv 10.200.134.25
iso.3.6.1.2.1.2.2.1.1.1 = INTEGER: 1
iso.3.6.1.2.1.2.2.1.1.2 = INTEGER: 2
iso.3.6.1.2.1.2.2.1.1.3 = INTEGER: 3
iso.3.6.1.2.1.2.2.1.1.4 = INTEGER: 4
iso.3.6.1.2.1.2.2.1.1.5 = INTEGER: 5
iso.3.6.1.2.1.2.2.1.1.6 = INTEGER: 6
iso.3.6.1.2.1.2.2.1.1.7 = INTEGER: 7
iso.3.6.1.2.1.2.2.1.1.8 = INTEGER: 8
iso.3.6.1.2.1.2.2.1.1.9 = INTEGER: 9
iso.3.6.1.2.1.2.2.1.1.10 = INTEGER: 10
iso.3.6.1.2.1.2.2.1.1.11 = INTEGER: 11
iso.3.6.1.2.1.2.2.1.1.12 = INTEGER: 12
iso.3.6.1.2.1.2.2.1.1.13 = INTEGER: 13
iso.3.6.1.2.1.2.2.1.1.14 = INTEGER: 14
iso.3.6.1.2.1.2.2.1.1.15 = INTEGER: 15
iso.3.6.1.2.1.2.2.1.1.16 = INTEGER: 16
iso.3.6.1.2.1.2.2.1.1.17 = INTEGER: 17
iso.3.6.1.2.1.2.2.1.1.18 = INTEGER: 18
iso.3.6.1.2.1.2.2.1.1.19 = INTEGER: 19
iso.3.6.1.2.1.2.2.1.1.20 = INTEGER: 20
iso.3.6.1.2.1.2.2.1.1.21 = INTEGER: 21
iso.3.6.1.2.1.2.2.1.1.22 = INTEGER: 22
iso.3.6.1.2.1.2.2.1.1.23 = INTEGER: 23
<...>
```

17. Security

17.1. Securing Management Plane

17.1.1. Securing the Management Plane Overview

The Securing Management Plane feature provides the capability to restrict the access to the management plane only to authenticated and authorized subjects.

The authentication identifies a subject, and the authorization validates if the subject is allowed to execute the action.

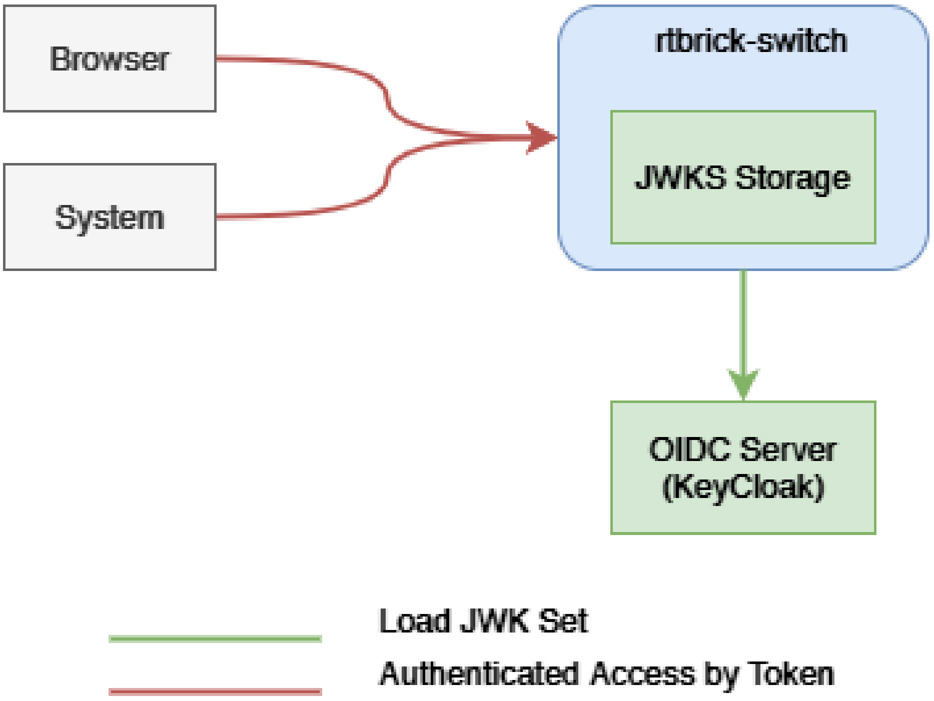


Figure 30. External Dataflow

The figure-1 shows the data flow when accessing an rtbrick-switch. Each call against the switch in more detail against the API Gateway Daemon (APIGWD) of the switch has to be authenticated with an access token. There is only one exception when accessing the CTLRD’s UI; it is possible to be redirected to an OpenIDConnect Authenticator.

The APIGWD validates the access token against an JSON Web Key Set (JWKS) (https://tools.ietf.org/html/rfc7517). This key set can be loaded from a file locally on the system or auto discovered via the OpenIDConnect server.

A valid access token, in the sense of syntactically correct but also successfully validated signature by one of the JSON Web Key of the JWKS files, leads in an authenticated user. If the validation is unsuccessful, the call will be rejected.

The access token contains scopes which are used internally for the authorization checks. The authorization is a role based authorization where the scopes equal to the roles.

Internally the access token is converted to an RtBrick token, and all the communications inside the switch is authenticated via this RtBrick token.

The dataflow inside of the switch can be seen in Figure 2.

The scopes of the access token are copied to the RtBrick Token.

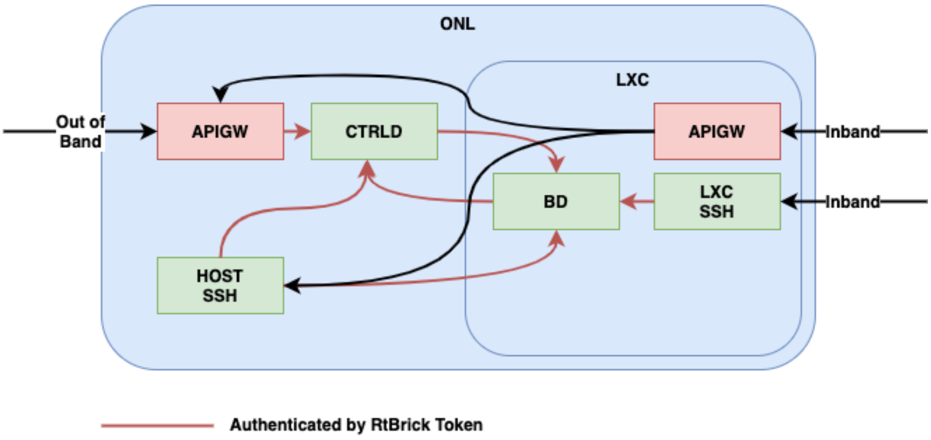


Figure 31. Internal Dataflow

Supported Platforms

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

17.1.2. RtBrick Token

JSON Web Tokens

JSON Web Token (JWT) is an open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object.

This information can be verified and trusted because it is digitally signed. JWTs can be signed using a secret (with the HMAC algorithm) or a public/private key pair using RSA or ECDSA.

RFC and draft compliance are partial except as specified.

For more information about JSON Web Token, see https://jwt.io/introduction/.

Structure

In its compact form, JSON Web Tokens consist of three parts separated by dots (.), which are:

- Header
- Payload
- Signature

Therefore, a JWT typically looks like the following.

```
xxxxx.yyyyy.zzzzz
```

Header

The header typically consists of two parts:

- The **type of the token**, which is JWT
- The signing algorithm that is being used, such as HMAC SHA256 or RSA

The suite of specifications on JWT provisions a few different options to identify particular cryptographic keys. The most straightforward mechanism is the "kid" claim. This claim can be added to the header of the token. It is intended to contain a string-based key identifier.

For example:

```
{
  "alg": "HS256",
  "typ": "JWT",
  "kid": "0815"
}
```

Then, this JSON is Base64Url encoded to form the first part of the JWT.

Payload

The second part of the token is the **payload**, which contains the claims. Claims are statements about an entity (typically, the user) and additional data.

There are three types of claims:

- registered
- public
- private

Registered claims

These are a set of predefined claims which are not mandatory but recommended, to provide a set of useful, interoperable claims. Some of them are: iss (issuer), exp (expiration time), sub (subject), aud (audience), and others.

Public claims

These can be defined at will by those using JWTs. But to avoid collisions they should be defined in the IANA JSON Web Token Registry or be defined as a URI that contains a collision resistant namespace.

Private claims

These are the custom claims created to share information between parties that agree on using them and are neither registered or public claims.

An example payload is as follows:

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "iat": 1516239022,
  "exp": 1600000000,
  "scope": "user"
}
```

The payload is then Base64Url encoded to form the second part of the JSON Web Token.

Signature

To create the signature part you have to take the encoded header, the encoded payload, a secret, the algorithm specified in the header, and sign that.

For example if you want to use the HMAC SHA256 algorithm, the signature will be created in the following way:

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  secret)
```

The signature is used to verify the message wasn’t changed along the way, and, in the case of tokens signed with a private key, it can also verify that the sender of the JWT is who it says it is.

Putting all together

The output is three Base64-URL strings separated by dots that can be easily passed in HTML and HTTP environments.

AccessToken

The Access Token is a JSON Web Token. The token is typically sent in the Authorization header using the Bearer schema. The content of the header should look like the following:

```
Authorization: Bearer <token>
```

The API Gateway also supports sending the token as Cookie, but this is not described here, that is only used for the CTRLD web UI.

The token has to have the kid claim in the header. This kid is used to find the right JSON Web Key (JWK) from one of the JSON Web Key Sets (JWKS).

APIGWD searches for the jwks file under /etc/rtbrick/apigwd/access_secret_jwks.json, but it is also possible to provide an additional oicd endpoint. By that the keysets are searched in the provided order:

- local file specified by command line -access-token-jwks-file-name
- oicd auto discovery -oidc-issuer

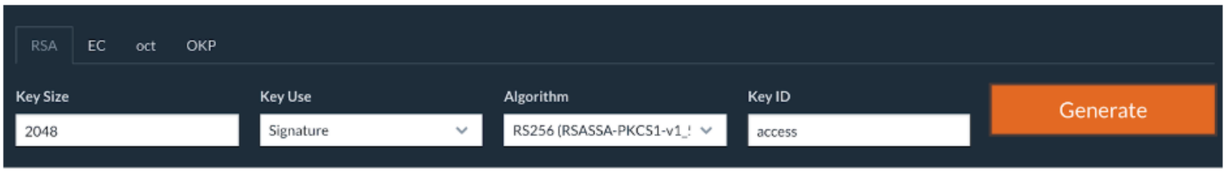
The scope claim contains the roles the user has. For example:

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "admin": true,
  "iat": 1516239022,
  "exp": 1600000000,
  "scope": "user operator"
}
```

This user has the roles user and operator.

JWKS Validation

The example below shows how to create a public/private key set file (for example with https://mkjwk.org/)



Key Size: 2048
Key Use: Signature
Algorithm: RS256 (RSASSA-PKCS1-v1_!
Key ID: access

Example PubPrivJwks.json:

```
{
  "keys": [
    {
      "p": "-
3WNqU2aWKy8Q7mblRpw_aOknW47YvZKVNQzlWdijXW5E1hQ5c6sfjqo92pq5mNKJ_Xkl71xHyp-WBfw-
xJqZ9pUuG4jnUiKgzYHvkccDF5XIMrpA67VnBozmyLckQOKEXesRD2hacrjb-
T89dcIZQHBUK1RYXGRxHCM_hPeBok",
      "kty": "RSA",
      "q": "07STzbuUh6p_iN2wzTegIQdnXBLnzPObCKt-KLSjkLtrZMv2YxM2yhMs-
56SsLR7EICFRAB2vdCzlovXKShubU9NSKgVI38qpGV9hii8rqN5N3dEM4Gscp_TR36ex1NoSC6kHBF2hsy
PfgD7U-txguZr6w9MN6rK4bAcg0TWGM0",
      "d": "IWcC4aKuPVlVGZeBhb3mla1mOsOcJuLxjpkZAu-
QaQO7phWzeGLEfln3tojKtIK6e11FEz137ow75Je0_oMAzE-
eTAMyseUTHZhn4LhmIdOwnsp9OZrMbNmLFpaF_rGYb0630xg27GdR4gC_lZvMuy1uCP1sr8Iyl1ujN7n_6
ETMYbdXPLOtEAB4Dag6XFTjy1l8aVvBxpscm8gKg64fgBGRvGY6PiUfqY3gaq_vX9SOOHVPN5WoShKA4fg
uxukRBiLLYNxDMDb3-h8pd1_fr2WayDzmcIXpxvVjRVZHt71C-
0Uhap6eRDMQocZXih8IdNV8zHUF_LeciE36fIb3oQ",
      "e": "AQAB",
      "use": "sig",
      "kid": "access",
      "qi": "3His_DaBkf_r7uDx9-8BOhOQPhcudT95XC9WyS5MrYIBtgqQi6IscHIqvtXFpjmPRey-
chO7p9msOAB_T8j_mg1l6UWOx6j4h_fyHEbOwRqfNemKng2Hs0uCrwpjgGf2eXzaBY8T9HlbFlTJAAARGh
_PePBi-F-IfAxGayj4hiM",
      "dp": "NJuYYpZAt1KUJJsdSKl6gCYPV3xrYj3iuTKYBCbYAH5jlP-
CFUIS5mnBVdnmuYKGTivsgi55DysluapwmSZ2KnoMBXXNb6dwixjvr8hSvuex1MN-
0m1udTUqHMfDW3dhGFxwJuq57VcsFAnVPl2ZfQBMAGPyRa-r7mwZo0Jmzfk",
      "alg": "RS256",
      "dq": "XL-
4IWIU6Hrh9OxrEP1VwiKkPcpqk3gGa_31_49kOXxiyH4zK6S3VECibHpEefYYFFq6B9jMLMzKYSJS2U1FU
85yZWp-GFcWL3_nRmeCgmBMMuilkIs3KeCrh58JoPoBrd4BN-rOqq_kDagQc-
uqh1a74PeKxLimucmWNExsH-E",
      "n": "z_NDmLu8M3KGvxvfJt8CAhdLdsqkskfY7vf9X9pW1LE_r31_HU85-
l6NNHeUWYbSNe6lt9YODnL8-
vTT6oCgre96byvpdYZ7Ki5KGe4fU96x0_ZF5LceUQc4l5dx6aptNi9mWgcZ9nkc2Xh83ASg9otG2YoYsAn
I1cO0TjzV9cMI7u7VON6SON9wbWFY01--ixMqxRAZuEJjbg4QAdL7DndRQXvmq1m7lv-
nnPPQ0a7ZTg7NZDEn5lMmadUlTVl5uvSNsACtC49R5kEkNCc1Hc-
3gootU5VyVPBx6IFHtNC2BiGasQAUpsDXZl7YtvBZwzYZwznUlluPiKLDk-4TtQ"
    }
  ]
}
```

The APIGWD only needs to know the Public part:

```
{
 "keys": [
 {
     "kty": "RSA",
     "e": "AQAB",
     "use": "sig",
     "kid": "access",
     "alg": "RS256",
     "n": "z_NDmLu8M3KGvxvfJt8CAhdLdsqkskfY7vf9X9pW1LE_r31_HU85-
l6NNHeUWYbSNe6lt9YODnL8-
vTT6oCgre96byvpdYZ7Ki5KGe4fU96x0_ZF5LceUQc4l5dx6aptNi9mWgcZ9nkc2Xh83ASg9otG2YoYsAn
I1cO0TjzV9cMI7u7VON6SON9wbWFY01--ixMqxRAZuEJjbg4QAdL7DndRQXvmq1m7lv-
nnPPQ0a7ZTg7NZDEn5lMmadUlTVl5uvSNsACtC49R5kEkNCc1Hc-
3gootU5VyVPBx6IFHtNC2BiGasQAUpsDXZl7YtvBZwzYZwznUlluPiKLDk-4TtQ"
   }
 ]
}
```

Now to create a token you can use https://keytool.online/, and paste the PubPrivJwks.json into the RSA Key Field and provide as Payload.

For example:

```
{
  "sub": "1234567890",
  "scope": "user operator",
  "name": "John Doe",
  "admin": true,
  "exp": 1600000000,
  "iat": 1516239022
}
```

This results in the following token:

```
eyJraWQiOiJhY2Nlc3MiLCJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdWIiOiIxMjM0NTY3ODkwI
iwic2NvcGUiOiJ1c2VyIG9wZXJhdG9yIiwibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWUsImV4cCI
6MTYwMDAwMDAwMCwiaWF0IjoxNTE2MjM5MDIyfQ.mP0mXR96-
9gYIzh6_2saUQckKDwpC7jDFpo2m0g9YAj4DkSf4xDoxBqMRwFkntLC6NV0sxyUNzC-
nv5yBretJAbbX_hCMS5Jk392piCVMt9ucbwnCKs6xaJDJmMHI1qxyf7lCgd9nIlawme4_nQnMJ4N9RdVeI
```

```
uyv1siuNUOo9RdSE4cX2JIzlrjgoZmtcU-
nq_I7S2QTkdro2e1wPZKktTMAoG6VjGb7ieIQ5XyLKNQt9PWSZ2sHkd85MxXMRUWUcrEagW6JrV3uixeT3
QTZ3g9Y6Qb4XDPH3EXUoAHJ0V26rpqXDsB_nmNvI5CVvCUcaZLPYoSEzBUPa9NaFIBcg
```

The apigwd can decode that token and validates the token with the corresponding key in the specified JKWS file.

OIDC Authentication

If you use OpenID Connect for Authentication, that the Token is generated by the OIDC Connect server.

It is important to understand how the validation of the tokens works. Either the JWKS file which corresponds to the OIDC server is located locally on the system, or the OpenID Connect Server (issuer) is specified.

The first configuration possibility we already discussed. If the oicd connect server is specified the server provides an endpoint where the clients can download the public keys.

As an example for this configuration of an oidc-issuer here an excerpt of:

/etc/rtbrick/apigwd/config.json

```
     "oidc_issuer" : "http://<keycloak>/auth/realms/<real name>",
     "client_id" : "<client id>",
     "client_secret" : "<secret>",
     "redirect_url": "",
```

Specific information about the issuer can be found at http://<keycloak>/auth/realms/<realm name>/.well-known/openid-configuration.

If you also specify the client secret and the client id, this allows the APIGWD to redirect to the login page of the OIDC server. This is needed for browser-based applications like CTRLD UI.

17.1.3. Role Based Access Control (RBAC)

Role Based Access Control (RBAC) is an approach to restrict the system access to authorized users. The authorization model is role-based. There will be three items in a role-based modeel: sub, obj, and act.

- **sub**: the user (role) that wants to access a resource.

- **obj**: the resource that is going to be accessed
- **act**: the operation that the user performs on the resource

The RBAC Data Model is implemented in RBFS, and it allows you to define Permission or User Roles to various type of resources.

The model contains:

- **Resource Type**: The type of resource we are talking about (for example, BDS Table, BDS Object, REST)
- **Resource**: The identifier of the Resource (for example, Table Name, Rest endpoints). Regular expressions are allowed.
- **Permissions**: Indicates the action that a user is allowed to perform on the resource. The Permissions are CRUD (Create, Read Update, Delete). The permission gets a semantic with respect to the resource type.
- **Role**: The role of a user who tries to access a resource.

CTRLD Authorization Configuration

Activate or Deactivate Authorization in CTRLD

```
"auth_disabled": true
```

It is possible to specify the permissions in CTRLD exactly in the way specified above.

Where sub is the role a user needs to have, obj species the url endpoint the user wants to reach, and act is the HTTP Method the user wants to call on the endpoint.

For example:

```
{
 "permissions": [
   {"sub": "supervisor", "obj": "/*", "act": ".*" },
   {"sub": "reader", "obj": "/*", "act": "GET"},
   {"sub": ".*", "obj":
"/api/v1/rbfs/elements/\{element_name}/services/\{service_name}/proxy/*", "act":
".*"}
 ]
}
```

- The user with the role supervisor is allowed to access all rest endpoints, and act on them with all HTTP methods.
- The user with the role reader is allowed to access all rest endpoints, but can only call the HTTP GET method.
- All authenticated users are allowed to access the proxy endpoint with all HTTP methods.

To configure that policy CTRLD offers 2 endpoints:

- PUT /api/v1/ctrld/authorization/permissions
- GET /api/v1/ctrld/authorization/permissions

Please refer to API Documentation for more information.

RBFS Authorization configuration

RBFS Role Configuration via REST

```
{
 "objects": [
   { "attribute": { "role": "operator", "permission": "create|read|delete",
"resource_regex": "global.*", "resource_type": "object" } },
   { "attribute": { "role": "operator", "permission": "create|read|delete",
"resource_regex": "global.*", "resource_type": "table" } }
 ],
 "table": { "table_name": "secure.global.rbac.authorization.config", "table_type":
"authorization_config_table" }
}

{
 "objects": [
   { "attribute": { "role": "user", "permission": "-|read|-", "resource_regex":
"global.*", "resource_type": "table" } },
   { "attribute": { "role": "user", "permission": "-|read|-", "resource_regex":
"global.*", "resource_type": "object" } }
 ],
 "table": { "table_name": "secure.global.rbac.authorization.config", "table_type":
"authorization_config_table" }
}
```

- **role** : Represents role in the system
- **resource_type** : Represents resources in the RBFS (table|object).
- **resource_regex** : Regex for the resources to be accessed.
- **permission** : Bitmap representing permissions to create, read and delete.

create|read|delete

Action	BDS Table	BDS Object
Create	Create a BDS Table	Create/Update a BDS Object
Read	Read Table Header Objects or Metadata	Read BDS Objects
Delete	Delete a BDS Object	Delete a BDS Object

RBFS Authorization CLI Configurations

Global user role configuration:

set system authorization global role <name> **rbac-permission** <resource-type> <resource-regex> **permission** <permission-map>

role	Represents role in the system
resource_type	Represents resources in the RBFS (table/object).
resource_regex	Regex for the resources to be accessed.
permission	Bitmap representing permissions to create, read and delete. -/-/- -/-/delete -/read/- -/read/delete create/-/- create/-/delete create/read/- create/read/delete

Example

```
admin@rtbick: cfg> set system authorization global role admin rbac-permission
table global.* permission create/read/delete
```

Lawful user role configuration

set system authorization lawful role <name> **rbac-permission** <resource-type> <resource-regex> **permission** <permission>

role	Represents lawful interceptor (LI) role in the system
resource_type	Represents resources in the RBFS (table/object).
resource_regex	Regex for the resources to be accessed.
permission	Bitmap representing permissions to create, read and delete. -/-/- -/-/delete -/read/- -/read/delete create/-/- create/-/delete create/read/- create/read/delete

Example

```
admin@rtbick: cfg> set system authorization lawful role fbi rbac-permission table
local.* permission -/read/-
```

17.1.4. SSH with TACACS+

RBFS provides a custom pluggable authentication module that gets invoked by the stock sshd on login. The necessary configurations are pre-installed on RBFS.

RtBrick-PAM, referred to as RTB-PAM helps in landing the TACACS authentication on the appropriate user in the Ubuntu container and helps in providing necessary details for the secure management plane feature.

Once the PAM client requests TACACS for the authentication, with successful authentication TACACS responds with a few RtBrick specific details.

```
{
     rtb-deny-cmds: “clear bgp peer”
     priv_lvl : some_level
}
```

On successful authentication, the RTB-PAM module creates a token (JWT) for the logged-in ssh user.

RTB-PAM Token

Token created by the RTB-PAM module contains the same claims that are defined under the RtBrick Token section, and this token is signed with the secret_jwks.json key.

The scope claim in the rtb-token is derived from the Linux groups that the locally mapped user belongs to.

The deny commands are converted into the claim rtb-deny-cmds. Once the token is created, it is transferred to the environment variable.

```
setenv RTB_TOKEN = {
  "sub": "83692",
  "iat": 1516239022,
  "exp": 1517239022,
  "name": "Admin User",
  "preferred_username", "user1",
  "scope": "operator tacacs_priv_lvl_8"
  "rtb-deny-cmds": "^clear bgp peer"
}
```

After the RTB-PAM token is created, the CLI prompt appears. If a token is not created for the logged-in user, then the user cannot perform communication with the BD.

SSH User Prompt

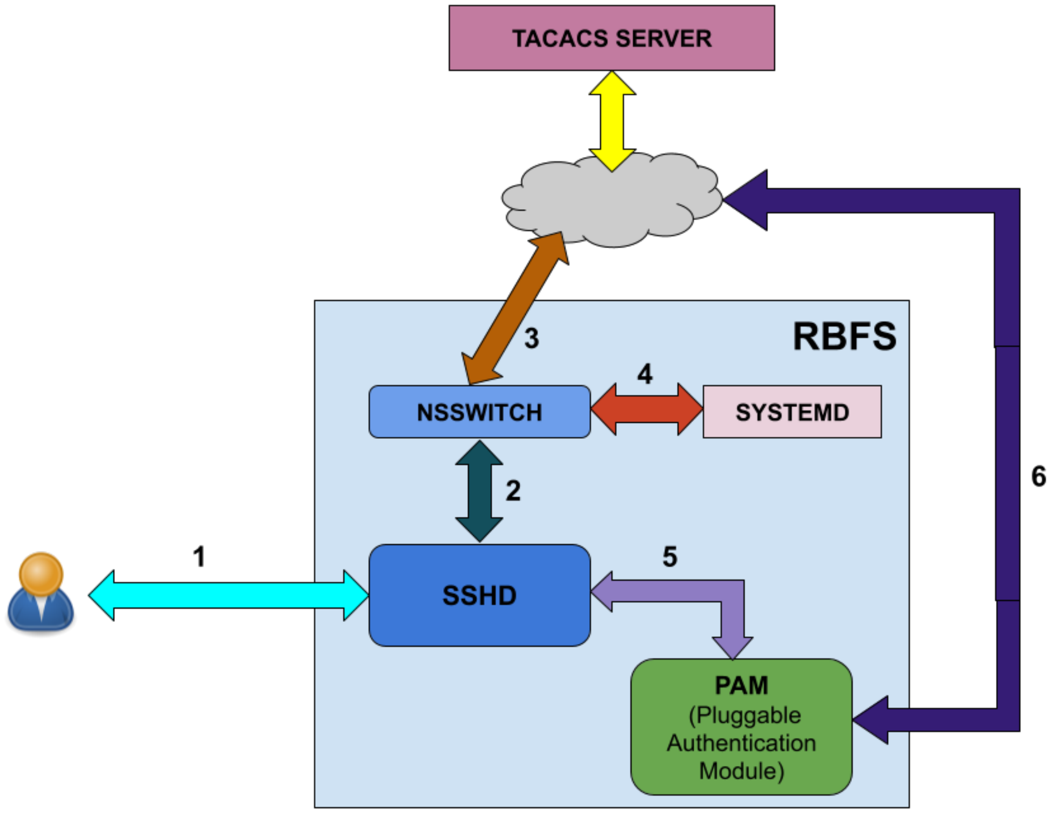
After you successfully log into RBFS via SSH, you can see the rtb-token using the shell environment. For example, an SSH prompt may look like the example below.

```
rtbng@b908f71f63b7:~$ env
SSH_CONNECTION=198.51.100.1 33136 198.51.100.44 22
LESSCLOSE=/usr/bin/lesspipe %s %s
LANG=C.UTF-8
USER=rtbng
PWD=/home/tacacs12
HOME=/home/tacacs12
SSH_CLIENT=198.51.100.1 33136 22
SUDO_USER=rtbng
PRIV_LVL=1
SSH_TTY=/dev/pts/1
SUDO_PROMPT=[sudo] password for rtbng:
MAIL=/var/mail/rtbng
TERM=xterm-256color
SHELL=/bin/bash
SHLVL=1
LOGNAME=rtbng
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/
local/games
LESSOPEN=| /usr/bin/lesspipe %s
_=/usr/bin/env
RTB_TOKEN=eyJhbGciOiJIUzI1NiIsImtpZCI6InJ0YnJpY2siLCJ0eXAiOiJKV1QifQ.eyJleHAiOjE0N
TE2MDczMDAsImlhdCI6MTQ1MTYwNjQwMCwiaXNzIjoicnRicmljay1hcGktZ3ciLCJuYW1lIjoiQmhpc2h
tYSBBY2hhcnlhIiwicHJlZmVycmVkX3VzZXJuYW1lIjoiYmhpc2htYSIsInNjb3BlIjoic3lzdGVtIiwic
3ViIjoiOTllOGI0YTEtM2E2Yi00YzI5LWJlZGItN2U3N2NjOTFjZTZiIn0.NOOAcafmHfgx-QFwiC-
_VGokbvUwrTOjhfpD9px3hMY
```

The users having access to the Linux shell only can see the installed rtb-token in the shell environment.

User Login Flow

The figure below shows the user login flow.



As shown in the figure above, the following steps are involved in the user login flow.

1. User starts to initialize using ssh
2. The SSH daemon (sshd) refers to the Name Service Switch (NSSWITCH) for user lookup
3. NSSWITCH performs a user lookup at the TACACS server

> A user lookup happens at TACACS user database even for the local users, except for the user 'supervisor'.

4. If the user is not found at TACACS user database, then the user look up happens at the local user database.
5. User authentication is performed by PAM modules
6. PAM communicates to the TACACS server for authentication if it is a TACACS user

Linux pre-configured users and groups

User Name	Group Name	Privilege
supervisor	supervisor	level 15
operator	operator	level 7-14
reader	reader	level 0-6

In-Band and Out-of-Band TACACS user SSH Login

In-band TACACS user SSH Login

A TACACS user can login using SSH to rtbrick container through inband management. RBFS should be configured with inband-management TACACS server for TACACS user login, and inband management configuration should be enabled with TACACS service.

Out-of-band TACACS user SSH Login

TACACS user can login via SSH to ONL though out-of-band management. RBFS should be configured with out-of-band management TACACS server for TACACS user login.

Configuring TACACS+ for RBFS

To configure TACACS+ server for RBFS, enter the the following commands.

Syntax

set system authorization tacacs server-ip <IP address> **type** <management type> **secret-plain-text** <secret key>

set system authorization tacacs server-ip <IP address> **type** <management type> **server-port** <server port number>

Command Arguments

<IP Address>	IP address of the TACACS Server
<management type>	in-band or out-of-band management

<IP Address>	IP address of the TACACS Server
secret-plain-text>	Secret plain text string. The secret string input can be plaintext format. If string starts with 1 then system considers it as encrypted string and stores key as it is. Also if secret string starts with 0, then system considers it as secret in plaintext and hence it stores in the system in encrypted format.
<server port number>	Server port number. This attribute is optional and by default system tries to connect to server running on port number 49.

For TACACS login to work with inband management, inband configuration must be enabled with the following command: **set inband-management instance** <instance-name> **tacacs true**

Example

```
root@rtbrick: cfg> set system authorization tacacs 198.51.100.111 out-of-band
secret-plain-text testkey

root@rtbrick: cfg> set system authorization tacacs server-ip 198.51.100.101 type
inband server-port 1234
```

A TACACS user is not allowed to login without TACACS server configuration.

The example below shows the running configuration after you configure TACACS.

```
{
  "rtbrick-config:system": {
    "authorization": {
      "tacacs": [
        {
          "ipv4-address": "198.51.100.111",
          "type": "out-of-band",
          "secret-encrypted-text": "$202a74ca845585855b6f8df57cdbf7858"
        }
      ]
    }
  }
}
```

Example: TACACS User Configuration in the TACACS Server

The example below shows the server configurations for rtb-deny-cmds.

```
accounting file = /var/log/tac_plus.acct
key = tacacskey

user = bob {
    login = cleartext "bob"
    member = Network_Operator
}
group = Network_Operator {
  default service = permit
  service = exec {
        priv-lvl = 10
        rtb-deny-cmds = "show bgp.*"
  }
}
```

> priv-lvl is a mandatory attribute in the TACACS user configuration.

Multiple cmd-regexes can be configured with each regexes separated by semicolon (;).

Example:

```
rtb-deny-cmds = "show bgp .*;show isis .*"
```

Troubleshooting NSS User Lookup Issues

To look up the TACACS username with all NSS methods, enter the following command:

```
ubuntu@rtbrick:~$ sudo getent passwd <tac_user>
```

To look up the local user within the local user database, enter the following command:

```
ubuntu@rtbrick:~$ sudo getent -s compat passwd <local_user>
```

To look up the TACACS user within the TACACS+ server database, enter the following command:

```
ubuntu@batman:~/development$ sudo getent -s tacplus passwd <tacuser>
```

If TACACS does not appear to be working correctly, You can enable debug logging by adding the debug=1 parameter to one or more of these files:

```
/etc/tacplus_servers
/etc/tacplus_nss.con
```

Configuring Secure Management Logs

SSH User Login Logs

The transaction logs of users (in the PAM module) are available in the following log file:

```
/var/log/auth.log
```

Commands to enable secure-management logging:

Full Syntax	set log bd <bd_name> module secure_management logmap <log_map> level <log-level>
Parameter descriptions	
Command modes	
Behavior description	Enables secure-management logging in the system
Example	confd> set log bd all module secure_management logmap all level Info

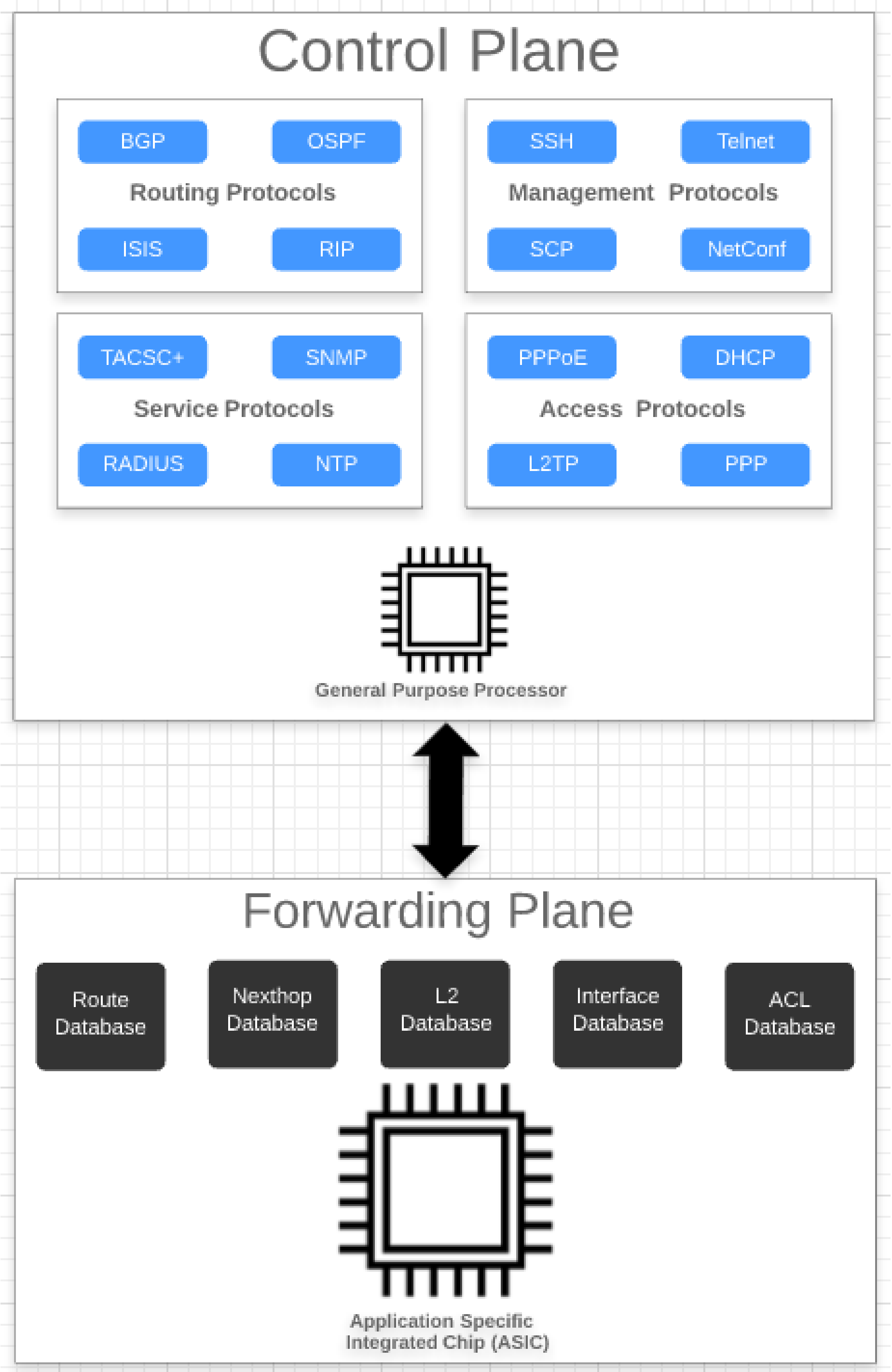
17.2. Securing Control Plane

17.2.1. Securing the Control Plane Overview

Control Plane Traffic

Control plane security enables you to filter or rate-limit unwanted traffic that is transmitted from the forwarding plane to the control plane. In RBFS, you can use Access Control Lists (ACL) and policers to secure the router’s control plane.

All routing protocols, management protocols, service protocols run in the control plane. The output of these protocols result in certain databases like routing table, MAC table, ARP table, etc., which eventually get programmed in the forwarding plane.

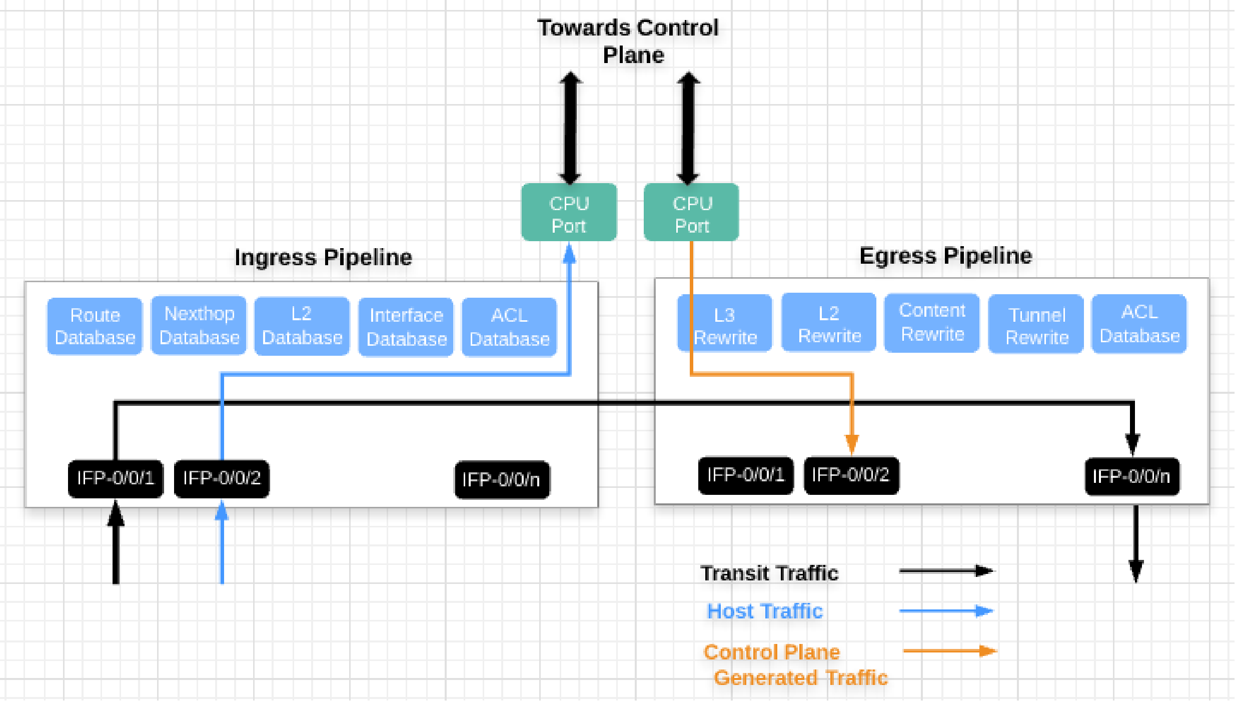


In the diagram above, you see the routing protocols (BGP, OSPF, ISIS), management protocols (SSH, RESTCONF, etc), service protocols (RADIUS, NTP, TACACS+), and access protocols (PPPoE, DHCP, L2TP, PPP) associated with the control plane. The control plane is generally implemented in software by using general-purpose processors. These protocols typically build a large number of databases like routing, switching, and ACL tables.

In contrast, a forwarding plane is associated with a copy of the databases (Routing, Switching, ACL, etc) built by the control plane. These entries typically contain the match and action which decide the packet flow in the forwarding plane. The forwarding plane functionality is realized in high performance Application Specific Integrated Circuits (ASICs) that are capable of handling very high packet rates.

There are two kinds of traffic:

1. **Control traffic**: Control traffic is destined to the device itself, that means, packets are handled by the router itself. The traffic is classified as control traffic based on matching destination IP, or because of ACL rules, or because of some kind of exception that occurred while parsing the packet (non-acceptable fields, TTL expiry, etc).
2. **Transit traffic**: Transit traffic not destined to the device itself. These packets will be sent out on one of the routers physical interfaces.



All control traffic packets will be destined to the CPU port(s). These packets are redirected to the control plane for further processing. However, general purpose processors in the control plane are not designed for packet processing, and might get overloaded if the rate of control plane traffic is too high, for example caused by a DDoS attack. Therefore you should to protect the router control plane by implementing mechanisms to filter completely or rate-limit traffic not required or unwanted at the control plane level.

Securing the Control Plane

In RBFS, there are two fundamental mechanisms how control-plane traffic is redirected to the CPU:

1. Via protocol ACLs
2. Via route lookup

Both mechanisms need to be considered and secured separately, as described in the following sections.

Control Plane Traffic via Protocol ACLs

All routing protocols (BGP, OSPF, ISIS), management protocols (SSH, RESTCONF, etc), service protocols (RADIUS, NTP, TACACS+), and access protocols (PPPoE, DHCP, L2TP, PPP), if enabled by configuration, automatically create Access Control Lists (ACLs) required to punt the protocol traffic to the control plane CPU. ACLs are the building block for securing the control plane. An ACL defines a rule, which typically contains match conditions and actions. If a packet matches the rule conditions, the associated actions will be applied. Protocol ACLs do not need to be defined by configuration. Another benefit is, they are very specific, for example match on auto-discovered IPv6 link-local neighbors. The protocol ACLs can verified using the 'show acl (detail)' command.

Example 1: Protocol ACL created by LLDP

```
supervisor@rtbrick: op> show acl detail
Rule: lldp.ifp-0/0/1.trap.rule
  ACL type: l2
  Ordinal: -
    Match:
      Attachment point: ifp-0/0/1
      Direction: ingress
      Destination MAC: 01:80:c2:00:00:0e
    Action:
      Redirect to CPU: True
    Result:
      Trap ID: LLDP
<...>
```

Example 2: Protocol ACL created by RADIUS

```
supervisor@rtbrick: op> show acl detail
Rule: radius-srv1-v4-auth-trap
  ACL type: l3v4
  Ordinal: -
    Match:
      Source L4 port: 1812
      IP protocol: UDP
    Action:
      Redirect to CPU: True
    Result:
      Trap ID: Radius
<...>
```

By default, for most of the control protocols, there is a single action Redirect to CPU: True. Thereby all traffic matching the match criteria gets punted to the CPU without any rate limit. There is one exception, for PPPoE only the traffic is rate-limited by default. As shown in the following example, there is an additional action Policer profile name: created by default that limits the PPPoE traffic to 50 Mbps per session:

Example 3: Protocol ACL with Policer created by PPPoE

```
supervisor@rtbrick: op> show acl detail
Rule: pppoed_hostif-0/0/1_7-7-1-4090_8863
  ACL type: PPPOE
  Ordinal: -
    Match:
      Attachment point: ifl-0/0/1
      Ethertype: 34915
    Action:
      Redirect to CPU: True
      Policer profile name: _DEFAULT_POLICER_50_MB
    Result:
      Trap ID: PPPoE
<...>
```

For all other protocols, rate limiting needs to enabled by configuration in order to secure the control plane. This is described in section 2.1 below.

Control Plane Traffic via Route Lookup

By default, any other traffic destined to one of the router’s IP addresses, commonly referred to as "my IP", and not matching any ACL is redirected to the CPU via a route lookup. This applies to loopback as well physical interface addresses. In order to secure the control plane against malicious traffic sent to one of the router’s IP addresses (not matching any protocol ACL), ACLs need to be defined by configuration. In RFBS, such "manually" created ACLs are referred to user-defined ACLs.

Typically you will want to completely block some unwanted traffic sent to "my IP", but allow and rate-limit some required traffic like for example ICMP. Please note, when designing the security ACLs to protect "my IP", you do NOT need to consider the protocol traffic already handled by the protocol ACLs.

When configuring ACLs, protocol ACLs and user-defined ACLs may conflict. For example, an ACLs created by the BGP routing protocol might match on TCP traffic sent to the routers loopback address with port 179. A user-defined ACL however might deny any traffic sent to this loopback address. In this case, protocol ACLs shall take precedence over user-defined ACLs, so that you do not accidentally break the protocol operation. In RBFS, this is implemented using different ACL database priorities.

Configuring ACLs to protect "my IP" is described in section 2.2 below.

Limitations and Notes

- The control-plane security features are supported on hardware platforms. They are not supported on virtual deployments.
- On the Edgecore AS5916-54XKS platform, BGPv6 with link-local peering uses route lookup instead of protocol ACLs. Therefore traffic sent to IPv6 link local addresses cannot be restricted via ACL to not break BGPv6 link-local peerings.

Supported Platforms

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

> Control Plane security is currently not supported for PIM, IGMP, and L2TP protocol traffic.

17.2.2. Control Plane Security Configuration

As highlighted above, there are two fundamental mechanisms how traffic is redirected to the CPU, via protocol ACLs, and via route lookup. These are addressed in following two sections.

Secure Control Plane Traffic via Protocol ACLs

This section describes the configuration options for control plane traffic that is redirected to the CPU via protocol ACLs. These ACLs are automatically created by the protocols, and do not need to be - and cannot be - configured manually. For example if you configure a routing protocol like BGP, the required ACLs to match and punt the BGP packets to the control plane are created automatically.

Enabling the Control Plane Security Feature

By default, all packets matching the protocol ACLs will be sent to the control plane without any rate limit, except for PPPoE. The RBFS Control Plane Security feature allows to add policers to all protocol ACLs. If enabled, this feature creates a set of default policers, and applies them to the protocol ACLs. Thereby the control plane gets secured against DDoS attacks matching these ACLs.

Syntax:

set forwarding-options control-plane-security <attribute> <value>

Attribute	Description
state (enable|disable)	Enable or disable the control-plane security feature. Default: disabled.

Example:

```
{
    "rtbrick-config:forwarding-options": {
      "control-plane-security": {
        "state": "enable"
      }
    }
}
```

Configuring Host Path QoS

The host-path-qos enable feature is disabled by default. Once it is enabled, you cannot disable it.

To enable the `host-path-qos' feature, enter the following command:

Syntax:

set forwarding-options class-of-service control-plane-qos ingress-qos <attribute> <value>

Attribute	Description
state (enable|disable)	Enable or disable the host path QoS feature. Default: disabled.

Example

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options class-of-service
control-plane-qos
{
    "rtbrick-config:control-plane-qos": {

      "ingress-qos": {
        "state": "enable"
      }
    }
  }
```

By enabling this feature, the default scheduler and queue configurations are installed and the CPU ports queue mapping will change. Also, all control plane ACLs will be reprogrammed to update action_forward_class.

Marking Outbound Control Plane Traffic

RBFS enables you to configure the various protocols to mark the egress control plane traffic. The control plane traffic can be marked with type-of-service (ToS) values.

- For BGP, OSPF, RADIUS, PIM, L2TPv2, and DHCP protocols, the remark-type should be configured as ToS
- For the IGMP and PPPoE protocols, the remark-type can be be configured as p-bit or tos
- The outbound-marking attributes such as name, code-point and remark-type are mandatory
- If the name of the protocol is L2-all, then the remark-type should configured as p-bit
- If the name of the protocol is L3-all, then the remark-type should configured as tos
- Redundancy related control traffic can be marked with specific tos value with the L3-all option

set forwarding-options class-of-service control-plane-qos outbound-marking protocol <protocol-name> <remark-type-value> **codepoint** <codepoint-value>

Option	Description
<protocol-name>	Specifies the protocol name.
<remark-type-value>	Specifies the remark type value that can be p-bit or tos.
<codepoint-value>	Specifies the codepoint value. The supported range for p-bit outbound-marking is 0-7.

Example: Marking Outbound CP Traffic Configuration

```
supervisor@rtbrick>LEAF01: cfg> show config
{
  "ietf-restconf:data": {
    "rtbrick-config:forwarding-options": {
      "class-of-service": {
        "control-plane-qos": {
          "outbound-marking": {
            "protocol": [
              {
                "protocol": "bgp",
                "remark-type": "tos",
                "codepoint": 192
              },
              {
                "protocol": "dhcp",
                "remark-type": "p-bit",
                "codepoint": 5
              },
              {
                "protocol": "l3-all",
                "remark-type": "tos",
                "codepoint": 224
              },
                    {
                "protocol": "igmp",
                "remark-type": "p-bit",
                "codepoint": 7
              },
              {
                "protocol": "igmp",
                "remark-type": "tos",
                "codepoint": 192
              },
              {
                "protocol": "ppp",
                "remark-type": "p-bit",
                "codepoint": 7
              },
              {
                "protocol": "ppp",
                "remark-type": "tos",
                "codepoint": 192
              },
              {
                "protocol": "radius",
                "remark-type": "tos",
                "codepoint": 100
            }
            ]
          }
        }
      }
    }
  }
}
```

Restricting Management Access

If you enable inband management access for example via SSH, protocol ACLs will be created that match on the enabled protocols and redirect the management traffic to the control plane. By default, this traffic is not restricted in terms of source IP addresses. You can optionally restrict management access to trusted IP addresses by applying a source prefix list. An additional match condition will then be added to the protocol ACLs for inband management.

Configuring a Prefix List

Syntax:

set forwarding-options prefix-list <options>

Option	Description
<prefix-list-name> ipv4-prefix <ipv4_prefix>	Prefix list configuration for IPv4.
<prefix-list-name> ipv6-prefix <ipv6_prefix>	Prefix list configuration for IPv6.

Applying a Prefix List

Syntax:

set inband-management instance <instance-name> source-prefix-list <list-name>

Example: Inband Management Configuration with Source Prefix List

```
"rtbrick-config:inband-management": {
      "instance": [
        {
          "name": "default",
          "ssh": "true",
          "ntp": "true",
          "source-prefix-list": "list1"
        }
      ]
    },

"rtbrick-config:forwarding-options": {
      "prefix-list": [
        {
          "prefix-list-name": "list1",
          "ipv4-prefix": [
            {
              "ipv4-prefix": "198.51.100.100/24"
            },
            {
              "ipv4-prefix": "198.51.100.33/24"
            },
            {
              "ipv4-prefix": "198.51.100.44/24"
            }
          ]
        }
```

ℹ The inband management is provided to only the source address specified in the prefix list if the prefix list is configured. If the prefix list not configured, it works for all source IPs. Also, the prefix addresses configured should be of /32.

Configuring Protocol ACL Options

This section describes how to configure policers per protocol and configure match on IPv4 ToS or IPv6 TC fields for protocol ACLs.

ℹ

- If control plane security is disabled, this configuration has no effect in the system
- Protocol-specific configuration will take priority over ALL configuration in the control-plane-security protocol

Syntax:

set forwarding-options control-plane-security protocol <protocol-name> <attribute> <value>

Option	Description
protocol <protocol-name>	Name of the protocol. You can configure individual protocols, and/or all protocols using the 'ALL' value keyword.
match-tc <tc-value>	Configure IPv6 TC value. The range is 0 to 248.
match-tos <tos-value>	Configure IPv4 ToS value. The range is 0 to 248.
policer <policer>	Configure policer name.

Example:

```
{
  "ietf-restconf:data": {
    "rtbrick-config:forwarding-options": {
      "class-of-service": {
        "policer": [
          {
            "policer-name": "_DEFAULT_POLICER_BGP_LL",
            "flags": "color-blind",
            "level1-rates": {
              "cir": 1000,
              "cbs": 1000,
              "pir": 1200,
              "pbs": 1000
            },
            "levels": 1,
            "type": "two-rate-three-color"
          }
        ]
      },
      "control-plane-security": {
        "state": "enable",
        "protocol": [
          {
            "protocol": "PIM",
            "policer": "_DEFAULT_POLICER_PIM",
            "match-tos": 192,
            "match-tc": 120
          }
        ]
      }
    }
  }
}
```

Secure Control Plane Traffic via Route Lookup

This section describes the configuration to secure the control plane for traffic that is redirected to the CPU via route lookup. Any packet sent to one of the router’s IP addresses ("my IP") and not matching any ACL, will be redirected to the CPU via route lookup. By default this type of traffic is not restricted or rate-limited. In order to secure the control plane, you need to apply ACLs by configuration. Please note you do not need to consider and allow any protocol traffic that is already captured by the automatically created protocol ACLs. You only need to explicitly define rules for any other traffic sent to "my IP". In the simplest case, you can deny any other traffic sent to the router. Typically you will want to allow some additional traffic like ICMP, and deny anything else.

Configuring ACLs

This section describes how to configure ACLs to secure the control plane to protect "my IP". In RFBS, ACLs are applied globally, that is, you do not need to attach them by configuration. Besides, for ACLs matching traffic sent to one of the router’s IP addresses, the redirect-to-cpu action applies implicitly and does not need to be configured.

Syntax:

set forwarding-options acl <options>

Option	Description
l3v4	ACL configuration for IPv4
l3v6	ACL configuration for IPv6
rule <rule-name>	Name of the ACL rule
ordinal <ordinal-value>	Number of the configuration entry. Please note the order of the configuration entries (ordinals) does not determine the processing.
match <condition>	Supported match conditions are IP source/destination prefix, prefix lists, source/destination Port, IP protocol, and direction.
action <action>	Supported actions are permit, drop, and police.

Option	Description
priority <value>	ACL entry priority. Determines the processing precedence for multiple matching i.e. conflicting rules. A less-specific rule should have a lower priority so that a more-specific rules takes precedence. Default: 10.

Example 1: Denying any Traffic destined to the Router’s Loopback Addresses

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options acl
{
    "rtbrick-config:acl": {
      "l3v4": {
        "rule": [
          {
            "rule-name": "BLOCK_LOOPBACK_TRAFFIC_v4",
            "ordinal": [
              {
                "ordinal-value": 20,
                "match": {
                  "destination-ipv4-prefix": "198.51.100.43/24",
                  "direction": "ingress"
                },
                "action": {
                  "drop": "true"
                },
                "priority": 20
              }
            ]
          }
        ]
      },
      "l3v6": {
        "rule": [
          {
            "rule-name": "BLOCK_LOOPBACK_TRAFFIC_v6",
            "ordinal": [
              {
                "ordinal-value": 20,
                "match": {
                  "destination-ipv6-prefix": "2001:db8:0:10::/32",
                  "direction": "ingress"
                },
                "action": {
                  "drop": "true"
                },
                "priority": 20
              }
            ]
          }
        ]
      }
    }
  }
supervisor@rtbrick>LEAF01: cfg>
```

Example 2: ACL allowing and rate-limiting ICMPv4/v6, and denying any other Traffic

```
supervisor@rtbrick>LEAF01: cfg> show config forwarding-options acl
{
    "rtbrick-config:acl": {
      "l3v4": {
        "rule": [
          {
            "rule-name": "BLOCK_LOOPBACK_TRAFFIC_v4",
            "ordinal": [
              {
                "ordinal-value": 20,
                "match": {
                  "destination-ipv4-prefix": "198.51.100.43/24",
                  "direction": "ingress",
                  "ip-protocol": "ICMP"
                },
                "action": {
                  "permit": "true"
                },
                "priority": 20
              }
            ]
          }
        ]
      },
      "l3v6": {
        "rule": [
          {
            "rule-name": "BLOCK_LOOPBACK_TRAFFIC_v6",
            "ordinal": [
              {
                "ordinal-value": 20,
                "match": {
                  "destination-ipv6-prefix": "2001:db8:0:10::/32",
                  "direction": "ingress",
                  "ip-protocol": "IPv6_ICMP"
                },
                "action": {
                  "permit": "true"
                },
                "priority": 20
              }
            ]
          }
        ]
      }
    }
  }
```

17.2.3. Control Plane Security Operational Commands

Show Commands

This section describes operational commands available to verify various control-plane security features.

Verifying ACLs

The show acl command allows to verify protocol ACLs as well as user-defined ACLs.

Syntax:

show acl <options>

Option	Description
detail	Displays all ACL details
<acl-name>	Displays the details for a single ACL

Example 1: Protocol ACL with Control-Plane Security enabled

```
supervisor@rtbrick>LEAF01: op>  show acl detail

Rule: lldp.ifp-0/0/1.trap.rule
  ACL type: l2
  Ordinal: -
    Match:
      Attachment point: ifp-0/0/1
      Direction: ingress
      Destination MAC: 01:80:c2:00:00:0e
    Action:
      Redirect to CPU: True
      Policer profile name: _DEFAULT_POLICER_50_MB
    Result:
      Trap ID: LLDP
<...>
Rule: radius-srv1-v4-auth-trap
  ACL type: l3v4
  Ordinal: -
    Match:
      Source L4 port: 1812
      IP protocol: UDP
    Action:
      Redirect to CPU: True
      Policer profile name: _DEFAULT_POLICER_20_MB
    Result:
      Trap ID: Radius
<...>
```

Example 2: ACL for Inband Management with Source Prefix List

```
supervisor@rtbrick>LEAF01: op>  show acl detail

Rule: ifm.inband.mgmt.lo-0/0/0/1.ssh.client.v4.trap.rule.1
  ACL type: l3v4
  Ordinal: 1
    Match:
      Destination IPv4 address: 198.51.100.91
      Source IPv4 address: 198.51.100.92
      Source L4 port: 22
      IP protocol: TCP
    Action:
      Redirect to CPU: True
    Result:
      Trap ID: INBAND
```

Example 3: User-defined ACL to Protect "my IP"

```
supervisor@rtbrick>LEAF01: op> show acl Protect-CP-v4

Rule: Protect-CP-v4
  ACL type: l3v4
  Ordinal: 1
    Match:
      Direction: ingress
      Destination IPv4 prefix: 198.51.100.91/24
      Source IPv4 prefix: 198.51.100.90/24
      IP protocol: ICMP
    Action:
      Permit: True
    Result:
      Trap ID: User Defined
  Ordinal: 2
    Match:
      Direction: ingress
      Destination IPv4 prefix: 198.51.100.91/24
    Action:
      Drop: True
    Priority: 5
    Result:
      Trap ID: User Defined
```

Verifying ACL Counters

The "show acl statistics" command displays information about the ACL packet counters. The counters are useful to verify if the ACL rules actually match, and if potentially malicious traffic gets dropped.

Syntax:

show acl statistics

Example 1: ACL statistics information

```
supervisor@rtbrick>LEAF01: cfg> show acl statistics
ACL                                                                            Units      Total
Accepted    Dropped
lldp.ifp-0/0/12.trap.rule                                                      Packets    -           -
```

```
-
                                                                                Bytes       -           -
-
lldp.ifp-0/0/16.trap.rule                                                       Packets     -           -
-
                                                                                Bytes       -           -
-
lldp.ifp-0/0/27.trap.rule                                                       Packets     -           -
-
                                                                                Bytes       -           -
-
lldp.ifp-0/0/53.trap.rule                                                       Packets     -           -
-
                                                                                Bytes       -           -
-
default_bgp_l4_trap_12::2_12::1_dst                                             Packets     12          12
0
                                                                                Bytes       1353        1353
0
default_bgp_l4_trap_12::2_12::1_src                                             Packets     12          12
0
                                                                                Bytes       1353        1353
0
default_bgp_l4_trap_12.0.0.2_12.0.0.1_dst                                       Packets     12          12
0
                                                                                Bytes       1353        1353
0
default_bgp_l4_trap_12.0.0.2_12.0.0.1_dst                                       Packets     -           -
-
                                                                                Bytes       -           -
-
default_bgp_l4_trap_12.0.0.2_12.0.0.1_src                                       Packets     12          12
0
                                                                                Bytes       1353        1353
0
default_bgp_l4_trap_12.0.0.2_12.0.0.1_src                                       Packets     -           -
-
                                                                                Bytes       -           -
-
supervisor@rtbrick: cfg>
```

Example 2: Display ACL statistics information for the specified ACL

```
supervisor@rtbrick>LEAF01: cfg> show acl default_bgp_l4_trap_12.0.0.2_12.0.0.1_dst statistics
ACL                                                                             Units       Total
Accepted    Dropped
default_bgp_l4_trap_12.0.0.2_12.0.0.1_dst                                       Packets     20          20
0
                                                                                Bytes       1917        1917
0
default_bgp_l4_trap_12.0.0.2_12.0.0.1_dst                                       Packets     -           -
-
                                                                                Bytes       -           -
-
supervisor@rtbrick>LEAF01: cfg>
```

Verifying Control Plane Policers

This command allows to view the policers created by the control-plane security feature.

Syntax:

show qos policer <options>

Option	Description
-	Displays all policers created by the control-plane security feature
<policer-name>	Displays information about the specified policer
counter	Displays all policer counters

Example 1: Display information of all policers created by the control-plane security feature

```
supervisor@rtbrick>LEAF01: cfg> show qos policer
Policer: _DEFAULT_POLICER_100_MB
Active: True, Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(KB)        PBS(KB)        Max CIR(Kbps)  Max PIR(Kbps)
  1        100000         100000         33000          33000          -              -
  2        -              -              -              -              -              -
  3        -              -              -              -              -              -
  4        -              -              -              -              -              -
Policer: _DEFAULT_POLICER_1_MB
Active: True, Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(KB)        PBS(KB)        Max CIR(Kbps)  Max PIR(Kbps)
  1        1000           1000           33000          33000          -              -
  2        -              -              -              -              -              -
  3        -              -              -              -              -              -
  4        -              -              -              -              -              -
Policer: _DEFAULT_POLICER_20_MB
Active: True, Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(KB)        PBS(KB)        Max CIR(Kbps)  Max PIR(Kbps)
  1        20000          20000          33000          33000          -              -
  2        -              -              -              -              -              -
  3        -              -              -              -              -              -
  4        -              -              -              -              -              -
Policer: _DEFAULT_POLICER_250_MB
Active: True, Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(KB)        PBS(KB)        Max CIR(Kbps)  Max PIR(Kbps)
  1        250000         250000         33000          33000          -              -
  2        -              -              -              -              -              -
  3        -              -              -              -              -              -
  4        -              -              -              -              -              -
Policer: _DEFAULT_POLICER_500_MB
Active: True, Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(KB)        PBS(KB)        Max CIR(Kbps)  Max PIR(Kbps)
  1        500000         500000         33000          33000          -              -
  2        -              -              -              -              -              -
  3        -              -              -              -              -              -
  4        -              -              -              -              -              -
Policer: _DEFAULT_POLICER_50_MB
Active: True, Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(KB)        PBS(KB)        Max CIR(Kbps)  Max PIR(Kbps)
  1        50000          50000          33000          33000          -              -
  2        -              -              -              -              -              -
  3        -              -              -              -              -              -
  4        -              -              -              -              -              -
Policer: _DEFAULT_POLICER_5_MB
Active: True, Type: two-rate-three-color, Levels: 1, Flags: -
  Level    CIR(Kbps)      PIR(Kbps)      CBS(KB)        PBS(KB)        Max CIR(Kbps)  Max PIR(Kbps)
  1        5000           5000           33000          33000          -              -
  2        -              -              -              -              -              -
  3        -              -              -              -              -              -
  4        -              -              -              -              -              -
supervisor@rtbrick>LEAF01: cfg>
```

Example 2: Display information of a specific policer

```
supervisor@rtbrick>LEAF01: cfg> show qos policer Premium_Upstream_Hierarchical_Policer
Policer: Premium_Upstream_Hierarchical_Policer
Active: False, Type: two-rate-three-color, Levels: 4, Flags: color-blind
  Level    CIR(Kbps)      PIR(Kbps)      CBS(KB)        PBS(KB)        Max CIR(Kbps)  Max PIR(Kbps)
  1        1000           1200           1000           1000           -              -
  2        900            1000           1000           1000           -              -
  3        5000           5200           1000           1000           -              -
  4        6000           6200           1000           1000           -              -
```

Example 3: Display information of policer counter

```
supervisor@rtbrick>LEAF01: cfg> show qos policer counter
Interface                         Level  Units      Total            Received         Dropped
ipv6_ll_prefix_acl                1      Packets    48               48               0
                                         Bytes      6383             6383             0

ipv6_mcast_ff01_prefix_acl        1      Packets    48               48               0
                                         Bytes      6383             6383             0

ipv6_mcast_ff02_prefix_acl        1      Packets    48               48               0
                                         Bytes      6383             6383             0

ppp-0/1/28/72339069014638594      1      Packets    0                0                0
                                         Bytes      0                0                0

ppp-0/1/28/72339069014638594      2      Packets    0                0                0
                                         Bytes      0                0                0

ppp-0/1/28/72339069014638594      3      Packets    0                0                0
                                         Bytes      0                0                0

ppp-0/1/28/72339069014638594      4      Packets    0                0                0
                                         Bytes      0                0                0

pppoed_ifp-0/1/28_1-3500-1-35     1      Packets    48               48               0
                                         Bytes      6383             6383             0

pppoed_ifp-0/1/28_1-3500-1-35     1      Packets    48               48               0
                                         Bytes      6383             6383             0

pppoed_ifp-0/1/30_1-3500-1-35     1      Packets    48               48               0
                                         Bytes      6383             6383             0

pppoed_ifp-0/1/30_1-3500-1-35     1      Packets    48               48               0
                                         Bytes      6383             6383             0
```

The show qos policer counter command displays the policer-level counters for the subscribers. The packets that get dropped after the RPF check, are currently updated in the local.bcm.q2c.trap.stats table in FIBD.

17.3. Local User Management

17.3.1. Local User Management Overview

Local User Management enables you to create, manage, and secure the Linux local users and groups through the RBFS configuration. This enables you to manage users and groups in the following environments:

- In the RBFS container only for the virtual platform
- In the RBFS container and on the ONL host for the hardware platforms

Supported Platforms

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

17.3.2. Local user management Configuration

RBFS allows you to create privileges that are configurable for user-defined and pre-defined roles. RBFS supports a combination of permit and deny regular expressions and a configurable default privilege to support both blacklisting and whitelisting of users. If both permit and deny command regular expressions match, the allow regular expression takes precedence.

Creating Roles

To create a role, you need to configure the following:

- Configure role-based access control (RBAC) privilege for the role
- Configure the command privilege for the role

> It is important to have secure management enabled when granting any type of privilege, whether it is RBAC or command-based, to any user. Without secure management, privileges do not work.

For information about enabling Secure Management, see the section "Configuring Secure Management Logs" of the *Securing Management Plane* user guide.

Configuring the RBAC Privilege

You need to configure the RBAC privilege for both table and object.

```
set system authorization global role <name> rbac-permission ( object | table ) <resource> <permission-type>
```

Command arguments

<name>	Authorization role name
<resource>	Represents resources in the RBFS (table/object)
<permission-type>	Permissions to create, read and delete. The following are the supported RBAC permission types: -/-/- -/-/delete -/read/- -/read/delete create/-/- create/-/delete create/read/- create/read/delete

Configuring the Command Privilege

```
set system authorization global role <name> cmd-permission ( allow-cmds <allow-cmds> | deny-cmds <deny-cmds> )
```

<role>	Authorization role name
<allow-cmds>	List of allow commands regular expression
<deny-cmds>	List of deny commands regular expression

- If you configure a privilege for any of the pre-defined roles (supervisor, operator, reader), then it replaces the default privilege.
- If you delete the configured privilege for a pre-defined role, then it will revert to the default privilege of the role.
- Priority of privilege rules is as follows: explicit deny, explicit permit, default privilege.

The example below shows the new role named "support" which has RBAC permission to read any table and objects. Also, the user is denied everything except the allowed commands (ping, set, show, traceroute, and watch-mode).

```
{
  "ietf-restconf:data": {
    "rtbrick-config:system": {
      "authorization": {
        "global": {
          "role": [
            {
              "name": "support",
              "rbac-permission": [
                {
                  "permission": "-/read/-",
                  "resource-type": "object",
                  "resource": ".*"
                },
                {
                  "permission": "-/read/-",
                  "resource-type": "table",
                  "resource": ".*"
                }
              ],
              "cmd-permission": {
                "allow-cmds": [
                  "ping .*",
                  "set .*",
                  "show .*",
                  "traceroute .*",
                  "watch .*"
                  ],
                "deny-cmds": ".*"
              }
            }
          ]
        }
      }
    }
  }
}
```

Linux pre-configured users, roles and privileges

User Name	Role Name	Default Privileges
supervisor	supervisor	Allow all actions
operator	operator	Allow all actions

User Name	Role Name	Default Privileges

reader	reader	All commands will be denied other than the commands which match any of the below regular expressions.
		`"color.*",`
		`"date.*",`
		`"exit.*",`
		`"history.*",`
		`"paging.*",`
		`"ping.*",`
		`"show.*",`
		`"traceroute.*",`
		`"watch-mode.*"`

Creating New Users

The new users created through local user management will always have a primary group with the same name and ID of the created user. The new user's ID will be allocated within the range of 3000 and 3999.

> You cannot use usernames such as root, wheel, admin, sudo or any of the SMP Linux pre-configured users and groups such as supervisor, operator, reader. Also, a username cannot start with "rtbrick_". If a Linux user with the same username already exists but has an ID outside of the 3000-3999 range then the user creation through the RBFS configuration will fail.

To create a new user, enter the following command:

```
set system user <username>
```

Command arguments

<username>	Name of the local user

Assigning Roles to Users

A "role" is an RBFS RBAC construct and it is mapped to a Linux group. The list of user roles from the RBFS configuration becomes the list of additional Linux groups that the Linux user belongs to. You can create new users and assign “roles” to the new users. The supervisor, operator, and reader are the pre-defined and pre-configured roles both in Linux and RBFS.

When a user is configured in RBFS under “system users”, RBFS/confd validates that the list of user roles only contain roles that are pre-defined or that are configured under “system authorization”.

> Do not create role names that start with “rtbrick_”. In addition, "root", "wheel", "admin", and "sudo" are not acceptable role names.

To assign a role to a new user, enter the following command:

```
set system user <username> role <role>
```

Command arguments

<username>	Name of the user
<role>	Role of the user (not the primary role)

Example: Assigning Roles to Users

```
{
  "ietf-restconf:data": {
    "rtbrick-config:system": {
      "user": [
        {
          "username": "bob",
          "role": [
            "operator"
            ],
          "shell": "/usr/local/bin/cli",
          "password-hashed-text":
"$6$uTE4OYn0iRq.Vppe$JBVMQ5DZHfuCuUP5yTnfl9IJsRLQAXqTLlLMKRO8bCz9WDlB2ele8puwMrT4/
QDF2nNOcoHtqYqFljly4B.Vu0"
        }
      ]
    }
  }
}
```

Configuring Authentication for a New User

There are two ways in which you can create a password for a new user in RBFS:

1. Configuring hashed password
2. Configuring plain-text password

Configuring Hashed Password

You can verify the integrity of your password using hashed passwords. When a user is present in the configuration but a "password hashed text" is not present, the password authentication is considered disabled for that specific user.

Three predefined roles include supervisor, operator, and reader. supervisor is the default password for the role supervisor. For the 'operator' and 'reader' roles, there is no default password. Use the system users supervisor command to disable the default supervisor password.

You can also disable password authentication for any of the predefined supervisor, operator and reader users by adding a "system users supervisor" configuration section without any "password hashed text" and thus disabling password authentication for the supervisor user.

SSH public keys can still be configured even if "password hashed text" is not present.

To create a password hashed text and authenticate the new user, perform the following steps:

1. Generate hash password on any Linux server.
   mkpasswd --method=SHA-512
2. Configure authentication using a password hashed text and an SSH public key.
   set system user <username> **password-hashed-text** <password-hashed-text>
   set system user <username> **ssh-pub-key** <ssh-pub-key>

Command arguments

<username>	Name of the user.

<password-hashed-text>	Password string.
<ssh-pub-key>	public keys of a user. You can specify multiple ssh-pub-keys.

3. Log in using username and password hashed text.

- It is possible to change shell, password-hash and ssh-pub-keys for supervisor, reader, and operator roles.
- The password string provided as part of the RBFS configuration needs to be a compatible password hashed text as defined by the shadow manual page: https://manpages.debian.org/buster/passwd/shadow.5.en.html and by the crypt https://manpages.debian.org/buster/manpages-dev/crypt.3.en.html.

Example

```
{
  "ietf-restconf:data": {
    "rtbrick-config:system": {
      "user": [
        {
          "username": "bob",
          "shell": "/usr/local/bin/cli",
          "password-hashed-text":
"$5$L2DaOYYuddhBV$9RA5MX9RQzLC9fIKJzbnoFBb88w9rkSXl7GVrVJ9PY7",
          "ssh-pub-key": [
            "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQCBAAABAQCubg5sdDycPN5EViNkV6w7rfp2GAfKWuInfaL3xOXyvSNpsmaHIL
YmgrLUU0GKQH9gauPUJpDcvvYaMt0ZBuTbWHVMUc4cvhgbNDkTB2bG2cTZ5QzbicyXff3BlDWQThVp2LtV
BiW2tf7JTTa9SnL4Lnm+CQcXsQ0rxqy2S6bJpsRYlFMyQl/hZ4QEWE153dw0HGvcG8mjfnPN4wvCc/omfD
3ljxx+Gf4oFS0davX6pdphUKLvgL33VVG5xaK71imv2l3897LIJZaHxy7FbB+CjSYT6QNq1XksX8omrbRj
iP3enEQi/bANtzTNnGDnIm1KHf3xuKpoKw+B5fhDZogx"
            ]
        }
      ]
    }
  }
}
```

Configuring a Plain-Text Password

To configure a password with plain text password for a new user, enter the following command:

set system user <username> **password-plain-text** <password-plain-text>

Command arguments

<username>	Name of the user
<password-plain-text>	Specifies the plain-text password

Example for Configuring Plain Text Password

```
set system user bob password-plain-text bob123
set system user bob shell /usr/local/bin/cli
set system user bob role operator
```

Viewing Configuration of Plain Text Password

```
{
  "ietf-restconf:data": {
    "rtbrick-config:system": {
      "user": [
        {
          "username": "bob",
          "role": [
            "operator"
            ],
          "shell": "/usr/local/bin/cli",
          "password-hashed-text":
"$6$uTE4OYn0iRq.Vppe$JBVMQ5DZHfuCuUP5yTnfl9IJsRLQAXqTLlLMKRO8bCz9WDlB2ele8puwMrT4/
QDF2nNOcoHtqYqFljly4B.Vu0"
        }
      ]
    }
  }
}
```

Setting the User Shell

RBFS validates that the shell is one of the following 3 valid options:

- /usr/sbin/nologin
- /bin/bash
- /usr/local/bin/cli

To configure user shell, enter the following command:

set system user <username> **shell** <shell>

Command arguments

<username>	Name of the user
<shell>	Name of the shell

Example

```
root@rtbrick: cfg> set system user smith shell /usr/local/bin/cli
```

Specifying the Display Name for User Names

The display name allows you to specify a preferred name so that you can easily identify the user. You can change your display name by entering the following command:

set system user <username> **display-name** <display_name>

Command arguments

<username>	Name of the user
<display_name>	Display name to easily identify the user

Example

```
set system user smith display-name primeuser
```

Enabling or disabling CLI access

You can control a user’s access to the CLI. By default, users will have access to the CLI.

set system user <username> **no-cli-access** < **true** | **false** >

Command arguments

<username>	Name of the user
<true | false>	When the no-cli-access is set to true, the user’s access to the CLI is disabled. When the no-cli-access is set to false, the user will be able access the CLI.

Example

```
set system user smith no-cli-access false
```

Configuring sudo Without Password

You can configure local system users to log in via passwords or using SSH keys. From a security perspective, it is desirable to allow authentication with SSH keys only. RBFS provides a configuration knob to disable the requirement for a 'sudo' password so that local users can authenticate with SSH keys only. This knob is configurable only if the user or one of its roles is supervisor.

You can enter the following command to enable or disable the 'sudo' password. By default, this is set to false which ensures that the supervisor must provide a password when using sudo.

set system user <user> **no-sudo-password** < **true** | **false** >

Command arguments

<username>	Name of the user
<true | false>	When the no-sudo-password is set to true, it indicates that a 'sudo' password is not required. When it is set to false, it indicates that the supervisor must provide a password when using sudo.

Example Configuration:

```
{
    "rtbrick-config:user": [
      {
        "username": "smith",
        "role": "supervisor",
        "shell": "/bin/bash",
        "ssh-pub-key": "ssh-rsa AAAAB3Nza<...>",
        "no-sudo-password": "true"
      }
    ]
 }
```

Note: If 'no-sudo-password' is set, you can log in with your SSH key.

Configuring Fail2Ban

The failed SSH login attempts from one user over an SSH jump host affect other users from the same jump host. You can configure Fail2Ban, which enables you to whitelist some IP addresses by creating separate jails for each user connecting through the jump host. In this way, failed login attempts by one user will not affect another user.

The following Fail2Ban command allows you to configure a list of IP addresses to include those IP addresses in the whitelist. You can specify multiple IP addresses that you want to exclude from the ban. Fail2Ban is applicable to both the ONL and the Linux container.

Syntax:

set system platform-management fail2ban ignore-ip <ignore-ip>

Command arguments

<ignore-ip>	Specify the IP addresses which are to be whitelisted.

Example commands for configuration:

```
supervisor@rtbrick: cfg> show config set
set system
set system platform-management fail2ban ignore-ip 10.1.1.1
set system platform-management fail2ban ignore-ip 10.1.1.2
set system platform-management fail2ban ignore-ip 10.1.1.3
```

Example Configuration:

```
supervisor@dev>rohit: cfg> show config
{
  "ietf-restconf:data": {
    "rtbrick-config:system": {
      "platform-management": {
        "fail2ban": {
          "ignore-ip": [
            "10.1.1.1",
            "10.1.1.2",
            "10.1.1.3"
            ]
        }
      }
    }
  }
}
```

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