



Forwarding Configuration Guide

Version 21.6.1, 25 June 2021

Registered Address	Support	Sales
26, Kingston Terrace, Princeton, New Jersey 08540, United States		
		+91 80 4850 5445
http://www.rtbrick.com	support@rtbrick.com	sales@rtbrick.com

©Copyright 2021 RtBrick, Inc. All rights reserved. The information contained herein is subject to change without notice. The trademarks, logos and service marks ("Marks") displayed in this documentation are the property of RtBrick in the United States and other countries. Use of the Marks are subject to RtBrick's Term of Use Policy, available at <https://www.rtbrick.com/privacy>. Use of marks belonging to other parties is for informational purposes only.

Table of Contents

1. Introduction to Forwarding	7
1.1. Guidelines and Restrictions	7
2. Interface Commands	8
2.1. Creating a Physical Interface	8
2.2. Creating a Logical Interface	8
2.3. Delete a Logical Interface	8
2.4. Assign Logical Interface IPv4 Address	9
2.5. Assign Logical Interface IPv6 Address	9
2.6. Delete Logical Interface IPv4 Address	10
2.7. Delete Logical Interface IPv6 Address	10
2.8. Specifying Interface Description	11
2.9. Setting the Admin Status of a Physical Interface	11
2.10. Disable (Shut Down) a Logical Interface	11
2.11. Enable a Logical Interface	12
2.12. Disable IPv4 on a Logical Interface	12
2.13. Disable IPv6 on a Logical Interface	13
2.14. Disable MPLS on a Logical Interface	13
2.15. Enable IPv4 on a Logical Interface	13
2.16. Enable IPv6 on a Logical Interface	14
2.17. Enable MPLS on a Logical Interface	14
2.18. Disable Physical Interfaces	14
2.19. Set IPv4, IPv6 or MPLS MTU Size	15
2.20. Configure an MPLS label with an IPv4 Address	16
2.21. Configure Interface Port Speed	16
2.22. Ping an IPv4 Address that is Part of a VRF Instance	17
2.23. Ping an IPv6 Address that is Part of a VRF Instance	17
2.24. Extended Community Support for Static and Interface Routes	18
2.24.1. Community setting for interface routes	18
2.25. Configuring Port Mirroring	18
2.25.1. Mirroring one physical interface traffic to another physical interface	19
2.25.2. Mirroring one logical interface traffic to another physical interface	20
2.25.3. Capturing Mirror Traffic	20
2.25.4. Mirror Logical Interface Traffic to CPU	21
2.25.5. Deleting Port Mirroring Configuration	22
2.26. Host Path Capturing Tool	22
2.26.1. Shared Memory Interface	22
2.26.2. Logical Interface	22
2.26.3. Physical Interface	22

2.26.4. Filtering by Protocol	23
2.26.5. Raw Option Capture tool	23
2.26.6. PCAP file format	23
2.27. IP Fragmentation	24
2.27.1. Guidelines and Limitations of IP Fragmentation	24
2.28. Firewall Filter Configuration	24
2.28.1. IPv4 Match Configuration (global)	25
2.28.2. IPv6 Match Configuration (global)	25
2.28.3. IPv4/IPv6 Action Configuration (global)	26
2.29. Prefix List for ACL and Multifield Classifier	27
2.29.1. Configuring IPv4 Prefix List for ACL and Multifield Classifier	27
2.29.2. Configuring IPv6 Prefix List for ACL and Multifield Classifier	27
2.29.2.1. Configuring Prefix List for ACL	28
2.29.3. Attaching Prefix List to ACL	28
2.29.3.1. Attaching an IPv4 Prefix List to ACL	28
2.29.3.2. Attaching an IPv6 Prefix List to ACL	28
2.29.4. Attaching Prefix List to Multifield-Classifer	29
2.29.4.1. Attaching an IPv4 Prefix List to Multifield-Classifer	29
2.29.4.2. Attaching an IPv6 Prefix List to Multifield-Classifer	30
3. Static Routing	31
3.1. Configure Static Route	31
3.2. Configuring nexthop-profile	32
3.3. Conditional Static Route	34
3.3.1. Conditional Profile Configuration	34
3.3.2. Conditional Static Route Configuration	35
3.4. Community Configuration	36
3.4.1. Extended Community configuration	36
3.5. Static Multicast Route Configuration	36
3.5.1. Viewing the Static Multicast Route Configuration	36
3.5.1.1. Delete Static route configuration	37
3.5.2. Deleting nexthop-profile	37
4. Configuring Link Aggregation Groups (LAGs)	39
4.1. Overview	39
4.1.1. L2 and L3 Interfaces	39
4.1.2. LACP (Link Aggregation Control Protocol)	39
4.2. Configuring Link Aggregation Groups (LAGs)	40
4.2.1. Creating LAG interface	40
4.2.2. Setting minimum-link-count on LAG interface	40
4.2.3. Setting member-interface on LAG interface	41
4.2.4. Setting LAC mode on LAG member interface	41
4.2.5. Setting timeout on LAG member interface	42

4.3. Show Commands	43
4.3.1. Displaying LAG running configuration	43
4.3.2. LAG show commands	43
4.3.2.1. Displaying LAG Summary	44
4.3.2.2. Displaying detailed information of LAG	44
4.3.2.3. Displaying detailed LAG information for a specific LAG interface	44
4.3.2.4. Displaying the specified LAG mode information for all LAG interfaces	45
4.3.2.5. Displaying information for the specified LAG mode and LAG interface	45
4.3.2.6. Displaying detailed information for the specified LAG mode and LAG interface	45
5. Show Commands	46
5.1. Instance Commands	46
5.1.1. show instance	46
5.1.2. show instance <instance-name>	46
5.1.3. show instance detail	46
5.1.4. show instance <instance-name> detail	47
5.2. Neighbor Commands	47
5.2.1. show neighbor	47
5.2.2. show neighbor <afi>	48
5.2.3. show neighbor <afi> instance <instance-name>	48
5.2.4. show neighbor instance <instance-name>	48
5.3. Address Resolution Commands	49
5.3.1. show address-resolution request	49
5.3.2. show address-resolution response	49
5.4. Route Show Commands	50
5.4.1. show route	50
5.4.2. show route detail	51
5.4.3. show route prefix <prefix/label>	51
5.4.4. show route prefix <prefix/label> detail	52
5.4.5. show route source <source>	52
5.4.6. show route source <source> detail	53
5.4.7. show route source <source> prefix <prefix>	53
5.4.8. show route source <source> prefix <prefix> detail	54
5.4.9. show route source <source> instance <instance-name>	54
5.4.10. show route source <source> instance <instance-name> detail	54
5.4.11. show route source <source> instance <instance-name> prefix <prefix>	55
5.4.12. show route source <source> instance <instance-name> prefix <prefix> detail	55

5.4.13. show route instance <instance-name>	56
5.4.14. show route instance <instance-name> detail	56
5.4.15. show route instance <instance-name> prefix <prefix>	57
5.4.16. show route instance <instance-name> prefix <prefix> detail	57
5.4.17. show route <afi>	58
5.4.18. show route <afi> detail	58
5.4.19. show route <afi> prefix <prefix>	59
5.4.20. show route <afi> prefix <prefix> detail	59
5.4.21. show route <afi> source <source>	60
5.4.22. show route <afi> source <source> detail	60
5.4.23. show route <afi> source <source> prefix <prefix>	61
5.4.24. show route <afi> source <source> prefix <prefix> detail	61
5.4.25. show route <afi> source <source> instance <instance-name>	62
5.4.26. show route <afi> source <source> instance <instance-name> detail	62
5.4.27. show route <afi> source <source> instance <instance-name> prefix <prefix>	63
5.4.28. show route <afi> source <source> instance <instance-name> prefix <prefix> detail	63
5.4.29. show route <afi> <safi>	64
5.4.30. show route <afi> <safi> detail	64
5.4.31. show route <afi> <safi> prefix <prefix>	65
5.4.32. show route <afi> <safi> source <source>	65
5.4.33. show route <afi> <safi> source <source> detail	66
5.4.34. show route <afi> <safi> source <source> prefix <prefix>	66
5.4.35. show route <afi> <safi> source <source> prefix <prefix> detail	66
5.4.36. show route <afi> <safi> source <source> instance <instance-name>	67
5.4.37. show route <afi> <safi> source <source> instance <instance-name> detail	67
5.4.38. show route <afi> <safi> source <source> instance <instance-name> prefix <prefix>	67
5.4.39. show route <afi> <safi> source <source> instance <instance-name> prefix <prefix> detail	68
6. ACL Show Commands	69
6.1. show acl	69
6.2. show acl detail	69
6.3. show acl <acl-rule-name>	70
7. ACL Statistics Commands	72
7.1. show acl statistics	72
7.2. show acl <acl-name> statistics	72
8. Interface Show Commands	73
8.1. show interface physical	73

8.2. show interface <interface-name>	75
8.3. show interface logical	75
8.4. show interface logical instance default	75
8.5. show interface address	76
8.6. show interface address instance default	76
8.7. show interface summary	76
8.8. show interface <interface-name>	78
8.9. show interface <interface-name> detail	78
8.10. show interface <interface-name> statistics	79
8.11. show interface <interface-name> statistics	85
8.12. show interface detail	87
8.13. show interface statistics	89
9. Statistics Show Commands	93
9.1. Viewing ACL Statistics	93
9.1.1. show bcm acl statistics	93
9.1.2. show bcm acl statistics type <type>	94
9.1.3. show bcm acl statistics type l3v4 rule <rule-name>	94
9.2. Viewing PPPOE statistics	95
9.2.1. show bcm pppoe counters	95
9.2.2. show bcm pppoe counters <subscriber-ID>	96
10. Clear Commands	97
10.1. Interface Counters	97
10.1.1. clear interface statistics	97
10.1.2. clear interface statistics <interface-name>	97
10.2. Clearing ACL Counters	97
10.2.1. clear acl counters	97
10.2.2. clear acl counters type <type>	97
10.2.3. clear acl counters type <type> rule <rule-name>	97
10.3. BCM clear Counter	97
10.3.1. Interface Counters	98
10.3.1.1. clear bcm interface statistics	98
10.3.1.2. clear bcm interface statistics <interface-name>	98

1. Introduction to Forwarding

FWDD interface commands fall into four major areas:

- Physical interface commands
- Logical interface commands
- Routing table (IPv4, IPv6, VRF, MPLS) commands

1.1. Guidelines and Restrictions

- The interface number displayed by RBFS may be different from the port number that you see on the front plate of the switch's chassis. This is because the interface number in RBFS always starts with 1, whereas the port number of the switch's front panel may start with either 0 to 1.

2. Interface Commands

2.1. Creating a Physical Interface

The following command creates a physical interface.

```
set interface <physical interface>
```

<physical interface>	Name of the physical interface
----------------------	--------------------------------

Example

```
root@rtbrick: cfg> set interface ifp-0/0/1
root@rtbrick: cfg> commit
```

2.2. Creating a Logical Interface



When an MPLS-labeled packet is double-tagged, no MPLS label termination is performed. Currently, RBFS supports single-tagged interfaces for MPLS.

The following command creates an interface on the default instance.

```
set interface <physical interface> unit <logical-unit-id> instance <instance>
```

<physical interface>	Name of the physical interface
<logical-unit-id>	Logical unit ID
<instance>	Name of the instance

Example

```
root@rtbrick: cfg> set interface ifp-0/0/1 unit 4071 instance ip2
root@rtbrick: cfg> commit
```

2.3. Delete a Logical Interface

Command to delete logical interface in instance (a show command will verify deletion)

delete interface <physical interface> **unit** <logical-unit-id>

<physical interface>	Name of the physical interface
<logical-unit-id>	Logical unit ID

Example

```
root@rtbrick: cfg> delete interface ifp-0/0/40 unit 4071
root@rtbrick: cfg> commit
```

2.4. Assign Logical Interface IPv4 Address

Command to assign an IPv4 address to a logical interface on the default instance and verify that the assignment is correct

set interface <physical interface> **unit** <logical-unit-id> **address ipv4** <ipv4-address>

set interface <physical interface> **unit** <logical-unit-id> **address ipv4** <ipv4-address> **secondary** <true | false>

<physical interface>	Name of the physical interface
<logical-unit-id>	Logical Unit ID
<ipv4-address>	IPv4 address to assign
<true false>	True or False

```
root@rtbrick: cfg> set interface ifp-0/0/1 unit 4071 address ipv4 10.10.10.1
root@rtbrick: cfg> set interface ifp-0/0/1 unit 4071 address ipv4 20.20.20.1 secondary
true
root@rtbrick: cfg> commit
```

2.5. Assign Logical Interface IPv6 Address

Command to assign an IPv6 address (as primary) to a logical interface on the default instance and verify that the assignment is correct

set interface <physical interface> **unit** <logical-unit-id> **address ipv6** <ipv6-address>

<physical interface>	Name of the physical interface
<logical-unit-id>	Logical Unit ID
<ipv6-address>	IPv6 address to assign

Example

```
root@rtbrick: cfg> set interface ifp-0/0/1 unit 4071 address ipv6 12:0:1::1/64
root@rtbrick: cfg> commit
```

2.6. Delete Logical Interface IPv4 Address

Command to delete the IPv4 address of a logical interface (without deleting the logical interface itself)

delete interface <physical interface> **unit** <logical-unit-id> **address ipv4** <ipv4-address>

<physical interface>	Name of the physical interface
<logical-unit-id>	Logical Unit ID
<ipv4-address>	IPv4 address

Example

```
root@rtbrick: cfg> delete interface lo-0/0/0 unit 4 address ipv4 4.4.4.4/32
```

2.7. Delete Logical Interface IPv6 Address

Command to delete the IPv6 address of a logical interface (without deleting the logical interface itself)

delete interface <physical interface> **unit** <logical-unit-id> **address ipv6** <ipv6-address>

<physical interface>	Name of the physical interface
<logical-unit-id>	Logical Unit ID
<ipv6-address>	IPv6 address

Example

```
root@rtbrick: cfg> delete interface lo-0/0/0 unit 4 address ipv6 4::4/128
```

2.8. Specifying Interface Description

You can specify a description for the interface by entering the following command:

```
set interface <physical interface> unit <logical-unit-id> description <description>
```

<physical interface>	Name of the physical interface
<logical-unit-id>	Logical Unit ID
<description>	Specifies the description for the interface

Example

```
root@rtbrick: cfg> set interface ifp-0/0/1 unit 4071 description "leaf1 to spine1"
root@rtbrick: cfg> commit
```

2.9. Setting the Admin Status of a Physical Interface

To set the admin status of a physical interface, enter the following command:

```
set interface <physical interface> admin-status <status>
```

<physical interface>	Name of the physical interface
<status>	Specifies the admin status, that is, up or down

Example

```
root@rtbrick: cfg> set interface ifp-0/0/1 admin-status down
root@rtbrick: cfg> commit
```

2.10. Disable (Shut Down) a Logical Interface

Command to disable (shut down) a logical interface on the default instance

set interface <physical interface> **unit** <logical-unit-id> **admin-status down**

<physical interface>	Physical interface name
<logical-unit-id>	Logical unit ID

Example

```
root@rtbrick: cfg> set interface ifp-0/0/52 unit 1 admin-status down
```

2.11. Enable a Logical Interface

Command to enable a logical interface that was previously shut down (essentially, delete the disable sent to the interface)

set interface <physical interface> **unit** <logical-unit-id> **admin-status up**

<physical interface>	Physical interface name
<logical-unit-id>	Logical Unit ID

Example

```
root@rtbrick: cfg> set interface ifp-0/0/52 unit 1 admin-status up
```

2.12. Disable IPv4 on a Logical Interface

Command to disable IPv4 on a logical interface on the default instance

set interface <physical interface> **unit** <logical-unit-id> **ipv4-admin-status down**

<physical interface>	Physical interface name
<logical-unit-id>	Logical Unit ID

Example

```
root@rtbrick: cfg> set interface ifp-0/0/52 unit 1 ipv4-admin-status down
```

2.13. Disable IPv6 on a Logical Interface

Command to disable IPv6 on a logical interface on the default instance

set interface <physical interface> **unit** <logical-unit-id> **ipv6-admin-status down**

<physical interface>	Physical interface name
<logical-unit-id>	Logical Unit ID

Example

```
root@rtbrick: cfg> set interface ifp-0/0/52 unit 1 ipv6-admin-status down
```

2.14. Disable MPLS on a Logical Interface

Command to disable MPLS on a logical interface on the default instance

set interface <physical interface> **unit** <logical-unit-id> **mpls-admin-status down**

<physical interface>	Physical interface name
<logical-unit-id>	Logical Unit ID

Example

```
root@rtbrick: cfg> set interface ifp-0/0/52 unit 1 mpls-admin-status down
```

2.15. Enable IPv4 on a Logical Interface

Command to enable IPv4 on a logical interface that was previously shut down.

set interface <physical interface> **unit** <logical-unit-id> **ipv4-admin-status up**

<physical interface>	Physical interface name
<logical-unit-id>	Logical Unit ID

Example

```
root@rtbrick: cfg> set interface ifp-0/0/52 unit 1 ipv4-admin-status up
```

2.16. Enable IPv6 on a Logical Interface

Command to enable IPv6 on a logical interface that was previously shut down.

set interface <physical interface> **unit** <logical-unit-id> **ipv6-admin-status up**

<physical interface>	Physical interface name
<logical-unit-id>	Logical Unit ID

Example

```
root@rtbrick: cfg> set interface ifp-0/0/52 unit 1 ipv6-admin-status up
```

2.17. Enable MPLS on a Logical Interface

Command to enable MPLS on a logical interface that was previously shut down.

set interface <physical interface> **unit** <logical-unit-id> **mpls-admin-status up**

<physical interface>	Physical interface name
<logical-unit-id>	Logical Unit ID

Example

```
root@rtbrick: cfg> set interface ifp-0/0/52 unit 1 mpls-admin-status up
```

2.18. Disable Physical Interfaces

You can disable all the unused physical interfaces.



- Always interface level enable/disable command has higher precedence than the global interface enable/disable command.
- Before executing global interface disable all command ensure that all physical interfaces are in the **link up** state.

To disable all the physical interfaces, enter the following command:

set global interface all admin-status down

The running configuration below shows the configuration changes.

```
{
  "data": {
    "rtbrick-config:global": {
      "interface": {
        "all": {
          "admin-status": "down"
        }
      }
    }
  }
}
```

Once global interface disable command is executed, all the interfaces will be down, and to enable a specific interface, you need to execute command below.

set interface <interface-name> admin-status up

Example

```
root@rtbrick: cfg> set interface ifp-0/0/52 admin-status up
```

If you want to enable all the physical interfaces that you disabled, enter the following command:

set global interface all admin-status up

2.19. Set IPv4, IPv6 or MPLS MTU Size

Command to assign an IPv4, IPv6 or MPLS MTU size to a logical interface

```
set interface <physical interface> unit <logical-unit-id> [ipv4-mtu <mtu-size> | ipv6-mtu <mtu-size> | mpls-mtu <mtu-size>]
```

<physical interface>	Physical interface name
<logical-unit-id>	Logical Unit ID
<ipv4-mtu>	Set MTU for IPv4
<ipv6-mtu>	Set MTU for IPv6
<mpls-mtu>	Set MTU for MPLS
<mtu-size>	Set the size of the MTU

Example

```

root@rtbrick: cfg> set interface ifp-0/0/1 unit 4071 ipv4-mtu 1500
root@rtbrick: cfg> set interface ifp-0/0/1 unit 4071 ipv6-mtu 1500
root@rtbrick: cfg> set interface ifp-0/0/1 unit 4071 mpls-mtu 1500
root@rtbrick: cfg> commit

```

2.20. Configure an MPLS label with an IPv4 Address

An MPLS label can be configured using the interface address configuration command. Enter the following command to configure a label with an IPv4 Address

set interface <interface-name> **unit** <logical-unit-id> **address** <ipv4> <address>
label <label>

<interface-name>	Name of the interface
<logical-unit-id>	Logical Unit ID
<label>	Specifies the interface label

Example

```

root@rtbrick: cfg> set interface lo-0/0/4 unit 1 address ipv4 192.1.4.3/32
label 12346

```

2.21. Configure Interface Port Speed

Port speed refers to the maximum amount of data transmitted through a port at any given second.

To configure port speed, enter the following command:

set interface <interface> **speed** <speed-value>

<interface>	Name of the physical interface
<speed-value>	Specifies the port speed, which is specified in Gigabits per second (Gbps)

Example

```

root@rtbrick: cfg> set interface ifp-0/0/1 speed 40G

```

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

2.22. Ping an IPv4 Address that is Part of a VRF Instance

Command to ping the IPv4 address used in the subscriber instance

ping <ipv4-address> **instance** <instance-name>

<ipv4-address>	IPv4 address to ping
<instance-name>	Name of instance to act on (for example, subscriber)

Example

```
supervisor@rtbrick: op> ping 192.1.4.3 instance subscriber
68 bytes from 192.1.4.3: icmp_seq=1 ttl=64 time=1.1126 ms
68 bytes from 192.1.4.3: icmp_seq=2 ttl=64 time=.0379 ms
68 bytes from 192.1.4.3: icmp_seq=3 ttl=64 time=.0399 ms
68 bytes from 192.1.4.3: icmp_seq=4 ttl=64 time=.0420 ms
68 bytes from 192.1.4.3: icmp_seq=5 ttl=64 time=.0391 ms
Statistics: 5 sent, 5 received, 0% packet loss
supervisor@rtbrick: op>
```

2.23. Ping an IPv6 Address that is Part of a VRF Instance

Command to ping the IPv6 address used in the subscriber instance

ping6 <ipv6-address> **instance** <instance-name>

<ipv6-address>	IPv6 address to ping
<instance-name>	Name of instance to act on (for example, subscriber)

Example

```
supervisor@rtbrick: op> ping6 192:1:4::3 instance subscriber
68 bytes from 192:1:4::3: icmp_seq=1 ttl=64 time=1.1199 ms
68 bytes from 192:1:4::3: icmp_seq=2 ttl=64 time=.0559 ms
68 bytes from 192:1:4::3: icmp_seq=3 ttl=64 time=.0404 ms
68 bytes from 192:1:4::3: icmp_seq=4 ttl=64 time=1.0944 ms
68 bytes from 192:1:4::3: icmp_seq=5 ttl=64 time=.0407 ms
Statistics: 5 sent, 5 received, 0% packet loss
supervisor@rtbrick: op>
```

2.24. Extended Community Support for Static and Interface Routes

After you configure the route and protocol policy, it is recommended to keep them generic using communities so that the future configurations or routes automatically get the desired processing.

Currently there are two items that cannot be "tagged" with a community but need to be operated by policy:

- static routes
- interfaces (interface routes)

The static routes provide a configuration option to add community/extended community.

For the interfaces, a community tag would be inherited to the interface routes (IPv4, IPv6 etc.) which is used by the policy while matching a community.

2.24.1. Community setting for interface routes

```
set interface ifp-0/0/52 unit 4 address ipv4 4.4.4.4/24 community
<community_value1>
set interface ifp-0/0/52 unit 4 address ipv4 4.4.4.4/24 community
<community_value2>

set interface ifp-0/0/52 unit 4 address ipv4 4.4.4.4/24 extended-community
<extended-community1>
set interface ifp-0/0/52 unit 4 address ipv4 4.4.4.4/24 extended-community
<extended-community2>
```

2.25. Configuring Port Mirroring

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.

Guidelines and Limitations

- You can configure up to 15 mirror profiles.
- The same mirror resources are used for 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 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.
- If physical interface/logical interface goes down, mirror configuration will not be deleted automatically. You need to delete the mirror configuration explicitly.

2.25.1. Mirroring one physical interface traffic to another physical interface

You can mirror one physical interface traffic to another physical interface by entering the following command:

set forwarding-options mirror <mirror-name> **destination interface** <interface-name>

set forwarding-options mirror <mirror-name> **source** <direction>

set forwarding-options mirror <mirror-name> **source interface** <interface-name>

<mirror_name>	Name for mirror configuration
<interface-name>	Port to where you need to mirror traffic
<direction>	Direction in which you want to mirror traffic (supported values ingress and egress)
<source-ifp-name>	Specifies the port from which you mirror the traffic

Example

```
supervisor@rtbrick: cfg> set forwarding-options mirror mm destination
interface ifp-0/0/4
supervisor@rtbrick: cfg> set forwarding-options mirror mm source direction
egress                ingress
supervisor@rtbrick: cfg> set forwarding-options mirror mm source direction
int
supervisor@rtbrick: cfg> set forwarding-options mirror mm source direction
ingress
supervisor@rtbrick: cfg> set forwarding-options mirror mm source interface
ifp-0/0/2
```

Example: Mirror Traffic to CPU

```
set forwarding-options mirror mirror1 destination interface cpu-0/0/200
set forwarding-options mirror mirror1 source direction ingress
set forwarding-options mirror mirror1 source interface ifp-0/0/52
```

2.25.2. Mirroring one logical interface traffic to another physical interface

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 is defined per Out-Port, or per Out-Port x VLAN tag. Configurations for seven distinct VLAN tags are supported.

Prerequisites

- Before creating logical interface mirroring, the source logical interface should exist.
- The logical interface should not be deleted during mirroring.

You can mirror one logical interface traffic to another physical interface by entering the following command:

set forwarding-options mirror <mirror-name> **destination interface** <interface-name>

set forwarding-options mirror <mirror-name> **source** <direction>

set forwarding-options mirror <mirror-name> **source logical-interface** <logical-interface>

<mirror_name>	Name for mirror configuration
<interface-name>	Port to where you need to mirror traffic
<direction>	Direction in which you want to mirror traffic (supported values ingress and egress)
<logical-interface>	Specifies the logical interface from which you are mirroring traffic

Example

```
set forwarding-options mirror mirror1 destination interface ifp-0/0/2
set forwarding-options mirror mirror1 source direction ingress
set forwarding-options mirror mirror1 source interface ifp-0/0/52
```

2.25.3. 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.

On the console, enter the following command:

```
show capture start mirrored
```

To capture the mirror traffic in a file, enter the following command:

```
show capture start mirrored file <filename>
```

Example

```
show capture start mirrored file test.pcap
```

To capture the mirror traffic on console, enter the following command:

```
show capture mirrored
```

2.25.4. Mirror Logical Interface Traffic to CPU

To mirror one logical interface traffic to CPU, enter the following command:

Syntax

```
set forwarding-options mirror <mirror-name> destination interface
<destination-cpu-interface-name>
set forwarding-options mirror <mirror-name> source <direction>
set forwarding-options mirror <mirror-name> source logical-interface
<logical-interface>
```

<mirror_name>	Name for mirror configuration
<destination-cpu-name>	CPU to where you need to mirror traffic
<source-ifl-name>	Specifies the logical interface from which you are mirroring traffic

Example

```
set forwarding-options mirror mirror1 destination interface cpu-0/0/200
set forwarding-options mirror mirror1 source direction ingress
set forwarding-options mirror mirror1 source interface ifp-0/0/52
```

2.25.5. Deleting Port Mirroring Configuration

You can delete the port mirroring configuration by entering the following command.

```
delete forwarding-options mirror [arg-required] <mirror_name>
```

Example

```
delete forwarding-options mirror mirror1
```

2.26. Host Path Capturing Tool

The sections below explain the various options available in the host path capturing tool to troubleshoot host path issues.

2.26.1. Shared Memory Interface

There is no bds packet tables in fibd, rather there is a pseudo network interface of the form shm-0/0/<trap-id>, where the trap-id identifies the protocol (BGP, ISIS, PPPoE, L2TP, RADIUS). You can use the VPP internal command **show rtb-shm** to know the mapping of protocol to trap-id. This command captures the packet exchanges between fibd and other protocol daemons.

```
show capture interface <shm-interface-name> direction <dir>
```

Example

```
show capture interface shm-0/0/1 direction both
```

2.26.2. Logical Interface

In most cases, when you want to troubleshoot, the packets are exchanged with other routers that are connected over a sub interface. The logical interface-level capturing enables this functionality.

Example

```
show capture interface ifl-0/0/52/1 direction both
```

2.26.3. Physical Interface

This is original physical interface-based capture that is supported. It is the entry

point to all packets transmitting to the software. If you want to know if packets are transmitting to software or not you can use this command.

Example

```
show capture interface ifp-0/0/52 direction both
```

2.26.4. Filtering by Protocol

In most cases, while using the logical interface and physical interface, you may want to select a packet belonging to a specific protocol. In that case, you can use the protocol knob available in all the commands to achieve this functionality.

Example

```
show capture interface ifp-0/0/52 direction both protocol bgp
show capture interface ifl-0/0/52/1 direction both protocol bgp
```

2.26.5. Raw Option Capture tool

The raw option capture tool can decode the packet as well as dump the packet in raw format. The **raw** option is useful if you want to examine packets in hex to check for malformed packets, etc.

Example

```
show capture interface ifl-0/0/52/1 direction both raw
show capture interface ifp-0/0/52 direction both raw
```

2.26.6. PCAP file format

While debugging a setup with real traffic, displaying the packets on console makes debugging tedious. You can use the **pcap** option to save the packets in the PCAP file and later run tcpdump/wireshark from any Linux system.

Example

```
show capture start interface ifp-0/0/52 direction both file test.pcap
show capture stop interface ifp-0/0/52 direction both file test.pcap
```


2.27. IP Fragmentation

If the maximum transmission unit (MTU) of an outgoing interface is lesser than the original packet which needs to be routed, the packet needs to be fragmented.

The current RBFS-supported Broadcom DNX chipset has no support for IP fragmentation in hardware. Because of this limitation, these packets are punted to the CPU, and the fragmentation is handled in 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 needs to perform fragmentation, the device sends an ICMP error code "Fragmentation needed and DF set" to the source.

2.27.1. Guidelines and Limitations of IP Fragmentation

The following guidelines and limitations are applicable to IP Fragmentation:

1. If a packet that's larger than 1492 which is negotiated with subscriber will get fragmented
2. The packets that are fragmented do not go over the regular QoS path in the egress pipeline
3. Broadcom supports at most 4 MTU profiles in a single device. This implies that there can only be 4 different kinds of MTU that can be set across all interfaces. With this limitation, the current interfaces can have any one of the following MTU profiles.
 - a. MTU 9216
 - b. MTU 1492
 - c. MTU 1518



By default, all the PPPoE interface will inherit the MTU 1492 profile.

2.28. Firewall Filter Configuration

Firewall filters provide a means of protecting your router (and switch) from excessive traffic transiting the router (and switch) to a network destination or destined for the Routing Engine. Also, firewall filtering restricts traffic destined for the Routing Engine based on its source, protocol, and application.

When you define a firewall filter, you define filtering criteria (terms, with match

conditions) for the packets and an action for the device to take if the packets match the filtering criteria.



The statistics for Firewall Filter/Ascend-Data-Filter is not supported because of a limitation in the supported hardware.

2.28.1. IPv4 Match Configuration (global)

set forwarding-options acl l3v4 rulename <rulename> **ordinal** <ordinal_value> **match** [**destination-ipv4-prefix** | **destination-l4-port** | **direction** | **ip-protocol** | **source-ipv4-prefix** | **source-l4-port** | **traffic-class**]

<rule_name>	Specifies the firewall filter rule name
<ordinal_value>	Specifies the ordinal value that is used for traffic policy rule referencing
<destination-ipv4-prefix>	Specifies the IPv4 prefix
<destination-l4-port>	Specifies the destination port number
<direction>	Ingress direction (Mandatory)
<ip-protocol>	Specifies the IP protocol
<source-ipv4-prefix>	Specifies the source IPv4 prefix
<source-l4-port>	Specifies the source IPv4 address
<traffic-class>	Specifies the traffic class value

Example

```
admin@rtbrick: cfg> set forwarding-options acl l3v4 rule rtb_firewall_two
ordinal 1000 match source-ipv4-prefix 40.1.1.0/24

admin@rtbrick: cfg> set forwarding-options acl l3v4 rule rtb_firewall_two
ordinal 1000 match source-l4-port 8080

admin@rtbrick: cfg> set forwarding-options acl l3v4 rule rtb_firewall_two
ordinal 1000 match direction ingress

admin@rtbrick: cfg> set forwarding-options acl l3v4 rule rtb_firewall_two
ordinal 1000 action drop true
```

2.28.2. IPv6 Match Configuration (global)

set forwarding-options acl l3v6 rulename <rulename> **ordinal** <ordinal_value> **match** [**destination-ipv6-prefix** | **destination-l4-port** | **direction** | **ip-protocol** | **source-ipv6-prefix** | **source-l4-port** | **traffic-class**]

<rule_name>	Specifies the firewall filter rule name
<ordinal_value>	Specifies the ordinal value that is used for traffic policy rule referencing
<destination-ipv6-prefix>	Specifies the IPv6 prefix
<destination-l4-port>	Specifies the destination port number
<direction>	Ingress direction (Mandatory)
<ip-protocol>	Specifies the IP protocol
<source-ipv6-prefix>	Specifies the source IPv6 prefix
<source-l4-port>	Specifies the source IPv4 address
<traffic-class>	Specifies the traffic class value

Example

```
admin@rtbrick: cfg> set forwarding-options acl l3v6 rule rtb_firewall_two
ordinal 1000 match direction ingress
admin@rtbrick: cfg> set forwarding-options acl l3v6 rule rtb_firewall_two
ordinal 1000 match source-ipv6-prefix 40::0/64
admin@rtbrick: cfg> set forwarding-options acl l3v6 rule rtb_firewall_two
ordinal 1000 match source-l4-port 8080
admin@rtbrick: cfg> set forwarding-options acl l3v6 rule rtb_firewall_two
ordinal 1000 action permit true
```

2.28.3. IPv4/IPv6 Action Configuration (global)

set forwarding-options acl [l3v4 | l3v6] rulename <rulename> ordinal <ordinal_value> action <action> [true | false]

<rule_name>	Specifies the firewall filter rule name
<ordinal_value>	Specifies the ordinal value that is used for traffic policy rule referencing
<action>	Specifies the filter action performed, that is, drop or permit
<true/false>	True indicates that the specified action is performed. False indicates that the specified action is not performed.

Example

```
admin@rtbrick: cfg> set forwarding-options acl l3v4 rule rtb_firewall_two
1000 action drop true
```

2.29. Prefix List for ACL and Multifield Classifier

A prefix list is a named list of prefixes. You can specify an exact match with incoming routes and apply a common action to all matching prefixes in the list. You can configure Access Control List (ACL) or Multifield Classifier (MFC) rules using prefix lists instead of individual prefixes.

Once both the Prefix List and the ACL using Prefix List is configured, the expanded ACL rules will be copied to the ACL Configuration table. This is different from existing behaviour where ACL rules are directly added to the ACL Configuration table.

- **Priority:** The priority will be copied from the prefix list ACL configuration in ACL candidate table to all the expanded ACL rules in ACL configuration table.
- **Ordinal:** A dedicated range (200001-4294967295) is reserved for the prefix list to be used for the expanded ACL rules in the ACL configuration table.

The validation process performs the following checks:

- You cannot configure the same prefix-list name to match the source prefix-list and destination prefix list.
- You cannot configure both source prefix and source prefix-list on the same ACL configuration.
- You cannot configure both destination prefix and destination prefix-list on the same ACL configuration.

2.29.1. Configuring IPv4 Prefix List for ACL and Multifield Classifier

set forwarding-options prefix-list <prefix-list-name> **ipv4-prefix** <ipv4_prefix>

<prefix-list-name>	Name of the prefix list which will be later used to attach with ACL configuration
<ipv4_prefix>	Specifies the IPv4 prefix address.

Example

```
admin@rtbrick: cfg> set forwarding-options prefix-list list1 ipv4-prefix
62.153.176.0/20
```

2.29.2. Configuring IPv6 Prefix List for ACL and Multifield Classifier

You can configure ACL or MFC rules using prefix lists instead of individual prefixes.

2.29.2.1. Configuring Prefix List for ACL

set forwarding-options prefix-list <prefix-list-name> **ipv6-prefix** <ipv6_prefix>

<prefix-list-name>	Name of the prefix list which will be later used to attach with ACL configuration
<ipv6_prefix>	Specifies the IPv6 prefix address.

Example

```
admin@rtbrick: cfg> set forwarding-options prefix-list ip6-list1 ipv6-prefix 10::10/128
```

2.29.3. Attaching Prefix List to ACL

2.29.3.1. Attaching an IPv4 Prefix List to ACL

set forwarding-options acl l3v4 rule <rule-name> **ordinal** <ordinal> **match source-ipv4-prefix-list** <prefix-list-name>

set forwarding-options acl l3v4 rule <rule-name> **ordinal** <ordinal> **match destination-ipv4-prefix-list** <prefix-list-name>

<rule-name>	Specifies the ACL rule name.
<ordinal>	Specifies the ordinal value that is used for traffic classification rule referencing.
<prefix-list-name>	Specifies the name of the source IPv4 prefix list.

Example

```
admin@rtbrick: cfg>set forwarding-options acl l3v4 rule rule1 ordinal 4 match source-ipv4-prefix-list list1
admin@rtbrick: cfg>set forwarding-options acl l3v4 rule rule1 ordinal 4 match destination-ipv4-prefix-list list2
```

2.29.3.2. Attaching an IPv6 Prefix List to ACL

set forwarding-options acl l3v6 rule <rule-name> **ordinal** <ordinal> **match source-ipv6-prefix-list** <prefix-list-name>

set forwarding-options acl l3v6 rule <rule-name> **ordinal** <ordinal> **match destination-ipv6-prefix-list** <prefix-list-name>

<rule-name>	Specifies the ACL rule name
<ordinal>	Specifies the ordinal value that is used for traffic classification rule referencing.
<prefix-list-name>	Specifies the name of the source/destination IPv6 prefix list.

Example

```
admin@rtbrick: cfg>set forwarding-options class-of-service multifield-
classifier acl l3v6 rule rule1 ordinal 4 match source-ipv6-prefix-list ip6-
list1
admin@rtbrick: cfg>set forwarding-options acl l3v6 rule rule1 ordinal 12
match destination-ipv6-prefix-list ip6_list2
```

2.29.4. Attaching Prefix List to Multifield-Classifier

2.29.4.1. Attaching an IPv4 Prefix List to Multifield-Classifier

set forwarding-options class-of-service multifield-classifier acl l3v4 rule <rule-name> **ordinal** <ordinal> **match source-ipv4-prefix-list** <prefix-list-name>

set forwarding-options class-of-service multifield-classifier acl l3v4 rule <rule-name> **ordinal** <ordinal> **match destination-ipv4-prefix-list** <prefix-list-name>



If you configure an ACL with prefix list that is not configured in the system or if the prefix list does not have any prefixes, then the ACL will not be configured.

<rule-name>	Specifies the multifield classifier rule name
<ordinal>	Specifies the ordinal value that is used for traffic policy rule referencing
<prefix-list-name>	Specifies the name of the source/destination IPv4 prefix list.

Example

```
admin@rtbrick: cfg>set forwarding-options class-of-service multifield-
classifier acl l3v4 rule rule2 ordinal 2 match source-ipv4-prefix-list list1
admin@rtbrick: cfg>set forwarding-options class-of-service multifield-
classifier acl l3v4 rule rule1 ordinal 4 match destination-ipv4-prefix-list
list2
```

2.29.4.2. Attaching an IPv6 Prefix List to Multifield-Classifier

set forwarding-options class-of-service multifield-classifier acl l3v6 rule <rule-name> **ordinal** <ordinal> **match source-ipv6-prefix-list** <prefix-list-name>

set forwarding-options class-of-service multifield-classifier acl l3v6 rule <rule-name> **ordinal** <ordinal> **match destination-ipv6-prefix-list** <prefix-list-name>



If you configure an ACL with prefix list that is not configured in the system or if the prefix list does not have any prefixes, then the ACL will not be configured.

<rule-name>	Specifies the multifield classifier rule name
<ordinal>	Specifies the ordinal value that is used for traffic policy rule referencing
<prefix-list-name>	Specifies the name of the source/destination IPv6 prefix list.

Example

```
admin@rtbrick: cfg>set forwarding-options class-of-service multifield-
classifier acl l3v6 rule rule2 ordinal 2 match destination-ipv6-prefix-list
ip6-list1
admin@rtbrick: cfg>set forwarding-options class-of-service multifield-
classifier acl l3v6 rule rule1 ordinal 4 match source-ipv6-prefix-list ip6-
list2
```

3. Static Routing

Static Routing allows a network administrator to configure routes manually. Using the RtBrick CLI, you can configure static ipv4, ipv6, MPLS and multicast routes.

3.1. Configure Static Route

To configure static route for ipv4/ipv6, enter the following command:

```
set instance <instance-name> static route <AFI> <prefix> <SAFI> <nexthop-profile>
```

To configure static route for MPLS, enter the following command:

```
set instance <instance-name> static route <AFI> <in-label> <in-bos> <SAFI> <nexthop-profile>
```

<instance-name>	Instance name to which destination prefix/label belongs to
<AFI>	Address family i.e ipv4, ipv6 and mpls can be given
<prefix>	Prefix address for which we wanted to install route
<SAFI>	Sub address family i.e unicast or labeled-unicast can be given
<in-label>	MPLS Label incoming CFG Key
<in-bos>	Incoming label BOS
<nexthop-profile>	Nexthop profile name where all nexthop related informaton like nexthop ip address, lookup-instance name, lookup-afi, lookup-safi are present.

Example

```
set instance default static route ipv4 10.1.1.1/32 unicast nexthop1
set instance default static route mpls 8888 true unicast nexthop1
set instance default static route ipv6 2001:db8:abcd:0012::0/80 unicast
nexthop2
```

Optional route parameters

Currently, you can configure below route parameters:

community	Community associated with the route
extended-community	Extended community associated with the route
swap-label	Label to be swapped

swap-bos	Label BOS
----------	-----------

3.2. Configuring nexthop-profile

You can group various nexthop parameters with nexthop-profile name and you can associate this nexthop-profile name with different routes.

Currently, the following nexthop parameters can be configured:

exit-interface	Exit interface Name
lookup-afi	Lookup routing tables address family where the nexthop will be looked up
lookup-instance	Lookup routing tables instance where the nexthop will be looked up
lookup-safi	Lookup routing tables sub address family where the nexthop will be looked up
nexthop	IPv4/IPv6 nexthop address
out-bos	Label BOS
out-label	Label that needs to be pushed

Syntax

```
set instance <instance-name> static nexthop-profile <profile-name>

set instance <instance-name> static nexthop-profile <profile-name> nexthop
<nexthop>

set instance <instance-name> static nexthop-profile <profile-name> exit-
interface <exit-interface>

set instance <instance-name> static nexthop-profile <profile-name> lookup-
afi <lookup-afi>

set instance <instance-name> static nexthop-profile <profile-name> lookup-
instance <lookup-instance>

set instance <instance-name> static nexthop-profile <profile-name> lookup-
safi <lookup-safi >

set instance <instance-name> static nexthop-profile <profile-name> out-
label <out-label>

set instance <instance-name> static nexthop-profile <profile-name> out-bos
<out-bos>
```

Example

nexthop-profile with ipv4 nexthop and out-label

```
set instance default static nexthop-profile nexthop1 nexthop 2.2.2.3
set instance default static nexthop-profile nexthop1 out-label 4444
```

nexthop-profile with ipv6 nexthop

```
set instance default static nexthop-profile nexthop2 nexthop fd3d:3d:100:a::2
```

nexthop-profile with null exit interface (discard route)

```
set instance default static nexthop-profile nexthop3 exit-interface null0
```

nexthop-profile with exit-interface

```
set instance default static nexthop-profile 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.

3.3. Conditional Static Route

By using the conditional Static Route configuration, you can make specific routes conditional. These conditional routes are installed into FIB if the specified condition is satisfied.

3.3.1. Conditional Profile Configuration

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 you can associate this conditional-profile name with different routes.

Syntax

```

set instance <instance-name> static conditional-profile <profile-name>

set instance <instance-name> static conditional-profile <profile-name>
match-instance <match-instance-name>

set instance <instance-name> static conditional-profile <profile-name>
match-afi <match-afi>

set instance <instance-name> static conditional-profile <profile-name>
match-safi <match-safi>

set instance <instance-name> static conditional-profile <profile-name>
compare-type <compare-type>

set instance <instance-name> static conditional-profile <profile-name>
compare-operation <compare-operation>

set instance <instance-name> static conditional-profile <profile-name>
compare-value <compare-value>

```

<instance-name>	Conditional routing compare operation.
<profile-name>	Conditional profile name
<match-instance-name>	Conditional routing tables instance where the condition will be checked

<match-afi>	Conditional routing tables address family where the condition will be checked
<match-safi>	Conditional routing tables sub address family where the condition will be checked
compare-type	Conditional routing compare type. The supported compare type is: route-count
<compare-operation>	Conditional routing compare operation. The only supported compare operation is greater-than . The less-than and equals compare operations are not supported.
compare-value	Conditional routing condition value

Example

```
set instance ip2vrf static conditional-profile c2
set instance ip2vrf static conditional-profile c2 match-instance default
set instance ip2vrf static conditional-profile c2 match-afi ipv4
set instance ip2vrf static conditional-profile c2 match-safi unicast
set instance ip2vrf static conditional-profile c2 compare-type route-count
set instance ip2vrf static conditional-profile c2 compare-operation greater-
than
set instance ip2vrf static conditional-profile c2 compare-value 20
```

3.3.2. Conditional Static Route Configuration

Syntax

```
set instance <instance-name> static route <afi> <prefix> <safi> <nexthop-
profile> conditional-profile <conditional-profile>
```

<instance-name>	Conditional routing compare operation.
<afi>	Conditional routing tables address family where the condition will be checked
<safi>	Conditional routing tables sub address family where the condition will be checked
prefix	Route prefix for which static route is installed
<nexthop-profile>	Nexthop profile name
<conditional-profile>	Conditional profile name

Example

```
set instance ip2vrf static route ipv4 0.0.0.0/0 unicast NULL conditional-
profile NULL_COUNT_10
```

3.4. Community Configuration

```
set instance default static route ipv4 10.1.1.1/32 unicast nexthop1 community
10:10
```

3.4.1. Extended Community configuration

```
set instance default static route ipv4 10.1.1.1/32 unicast nexthop1 extended-
community target:15:1
```

3.5. Static Multicast Route Configuration

syntax

```
set instance <instance-name> static route multicast4 <source> <group>
<nexthop-profile>
```

<instance-name>	Instance name to which source belongs to
<source>	IPv4 Multicast stream source address
<group>	IPv4 Multicast group address
<nexthop-profile>	Nexthop profile name where all nexthop related informaton like nexthop ip address,lookup-instance name, lookup-afi, lookup-safi will present.

Example

```
set instance default static route multicast4 10.1.1.1/32 232.1.1.1/32
nexthop3
```

3.5.1. Viewing the Static Multicast Route Configuration

```
show config instance <instance-name> static
```

Example

```
show config instance default static
```

3.5.1.1. Delete Static route configuration

To delete Static route configuration of ipv4/ipv6, enter the following command:

```
delete instance <instance-name> static route <AFI> <prefix> <SAFI>  
<nexthop-profile>
```

To delete Static route configuration of MPLS, enter the following command:

```
delete instance <instance-name> static route <AFI> <in-label> <in-bos>  
<SAFI> <nexthop-profile>
```

Example

```
delete instance default static route ipv4 10.1.1.1/32 unicast nexthop1  
delete instance default static route ipv6 2001:db8:abcd:0012::0/80 unicast  
nexthop2  
delete instance default static route mpls 8888 true unicast nexthop2
```

To delete Static route configuration of multicast, enter the following command:

```
delete instance <instance-name> static route multicast4 <source>  
<group> <nexthop-profile>
```

Example

```
delete instance default static route multicast4 10.1.1.1/32 232.1.1.1/32  
nexthop3
```

3.5.2. Deleting nexthop-profile

```
delete instance <instance-name> static nexthop-profile <profile-name>
```

Example

```
delete instance default static nexthop-profile nexthop1
```

4. Configuring Link Aggregation Groups (LAGs)

4.1. Overview

Link aggregation is a way of bundling or grouping multiple physical links to 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.

A LAG is a communication link between two switches. Multiple physical ports can be configured to form a single logical interface. This logical interface could be formed statically or dynamically. LACP protocol helps to bring up the interface dynamically. The LAG interface combines the bandwidth of the individual member links. The properties like speed, bandwidth of the individual member links should be the same to make it part of that LAG. The traffic which is directed towards the LAG interface is sent on the individual member links. This traffic is not pinned to a specific member link but rather determined by a specific flow. This hash could be calculated based on various fields in the packet.

LAG interfaces can be used as Layer 2 and Layer 3 interfaces. The configuration options which are available on the physical interface should also be available on the LAG interfaces.

4.1.1. L2 and L3 Interfaces

A regular L2 or L3 interface can be created on top of the single LAG interface. These interfaces can be divided based on 802.1q VLAN ID's. Multiple L3 interfaces can be created and each of them can be associated with different VRFs.

4.1.2. 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. Two kinds of LAG are supported.

1. **Static LAG:** In this mode, the member links do not initiate nor process any of the LACP packets received. This implies, the device brings up the LAG interface without LACP negotiation.

2. **Dynamic LAG:** In this mode, the member links processes 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 are generated packets on the member link in response to the LACP packet received, the receive side. That means, at least one side of the LAG should be configured as ACTIVE to bring the LAG interface.

4.2. Configuring Link Aggregation Groups (LAGs)

4.2.1. Creating LAG interface

When you create a lag interface in LACP mode, LACP PDUs are sent and received through member interfaces. Based LACP protocol state machine member port involves in traffic forwarding

Syntax:

```
set link-aggregation interface <lag-interface> mode <mode>
```

Attribute	Description
<member-interface>	Name of LAG interface.
<mode>	Specifies the LAG mode. The possible modes are: <ul style="list-style-type: none">• lacp - In this mode, the member links processes LACP packets received.• static: In this mode, the member links do not initiate or process any of the LACP packets received.

Example:

```
supervisor@rtbrick: cfg> set link-aggregation interface lag-4 mode lacp
```

4.2.2. Setting minimum-link-count on LAG interface

You can configure the minimum number of member links that must be up for the bundle as a whole to be labeled up. By default, only one link must be up for the bundle to be labeled up.

Syntax:

```
set link-aggregation interface <lag-interface> minimum-link-count
<minimum-link-count>
```

Attribute	Description
<lag-interface>	Name of LAG interface.
<minimum-link-count>	Specifies the minimum member links to be UP to declare LAG to UP. The default is 1.

```
supervisor@dev1: cfg> set link-aggregation interface lag-4 minimum-link-count
4
```

4.2.3. Setting member-interface on LAG interface

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.

Syntax:

```
set link-aggregation interface <lag-interface> member-interface
<member-interface>
```

Attribute	Description
<lag-interface>	Name of LAG interface.
<member-interface>	Name of LAG member interface.

Example:

```
supervisor@dev1: cfg> set link-aggregation interface lag-4 member-interface
ifp-0/0/1
```

4.2.4. Setting LAC mode on LAG member interface

You can set the LACP mode of the member interface. The two options available are:

- **ACTIVE:** In ACTIVE mode LACP runs on interface and the port automatically initiates negotiations with other ports by initiating LACP packets.

- **PASSIVE:** In PASSIVE mode LACP runs on and the port automatically not initiates negotiations with other ports with LACP packets, but ports in passive mode respond to LACP packets.

Syntax:

```
set link-aggregation interface <lag-interface> member-interface
<member-interface> lACP-mode <mode>
```

Attribute	Description
<lag-interface>	Name of LAG interface.
<member-interface>	Name of LAG member interface.
<mode>	Specifies the LACP mode. 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.

Example:

```
supervisor@dev1: cfg> set link-aggregation interface lag-4 member-interface
ifp-0/0/1 lACP-mode active
```

4.2.5. Setting timeout on LAG member interface

You can set the timeout value of the lag member interface. The two options available are:

- **long:** Ports with LONG timeout value use long timeouts to detect status of the aggregate, that means, they expect less frequent transmissions. Long timeout is 90 seconds.
- **short:** Ports with SHORT timeout wish to use short timeouts, that means, they expect frequent transmissions to detect status changes. Short timeout is 3 seconds.

Syntax:

```
set link-aggregation interface <lag-interface> member-interface
<member-interface> lACP-timeout <timeout-value>
```

Attribute	Description
<lag-interface>	Name of LAG interface.
<member-interface>	Name of LAG member interface.
<timeout-value>	Specifies the timeout for the LACP session. The default long time-out value is 90 seconds; short is 3 seconds.

Example:

```
supervisor@dev1: cfg> set link-aggregation interface lag-4 member-interface
ifp-0/0/1 lacp-timeout long
```

4.3. Show Commands

4.3.1. Displaying LAG running configuration

Enter the below command to see LAG configuration on the system.

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

4.3.2. LAG show commands

4.3.2.1. Displaying LAG Summary

To display the LAG interface summary, enter the **show lag** command.

```
supervisor@rtbrick: op> show lag
Lag-Interface      Member-Interface  actor-key partner-key
lag-4              ifp-0/0/1         10         10
                  ifp-0/0/4         10         10
```

4.3.2.2. Displaying detailed information of LAG

To display the details of all LAG interfaces, enter the **show lag detail** command.

```
supervisor@rtbrick: op> show lag detail
Lag-Interface : lag-4
Member-Interface : ifp-0/0/1
Actor
  system: 7a:67:1e:62:02:02
  system priority: 65535
  key: 15
  port priority: 255
  port number: 1
Partner
  system: 7a:37:f7:62:02:02
  system priority: 65535
  key: 15
  port priority: 255
  port number: 1
Member-Interface : ifp-0/0/4
Actor
  system: 7a:67:1e:62:02:02
  system priority: 65535
  key: 15
  port priority: 255
  port number: 2
Partner
  system: 7a:37:f7:62:02:02
  system priority: 65535
  key: 15
  port priority: 255
  port number: 1
```

4.3.2.3. Displaying detailed LAG information for a specific LAG interface

To display the details of a specific LAG interfaces, enter the **show lag <lag-name> detail** command.

```
supervisor@rtbrick: op> show lag lag-4 detail
```

4.3.2.4. Displaying the specified LAG mode information for all LAG interfaces

To display static or LACP information for all LAG interfaces, enter the **show lag mode <static|lcap>** command.

```
supervisor@rtbrick: op> show lag mode lcap
```

4.3.2.5. Displaying information for the specified LAG mode and LAG interface

To display static or LACP information for a specific LAG interface, enter the *** show lag mode <static|lcap> <lag-name>** command.

```
supervisor@rtbrick: op> show lag mode lcap lag-4
```

4.3.2.6. Displaying detailed information for the specified LAG mode and LAG interface

To display static or LACP information for a specific LAG interface, enter the **show lag mode <static|lcap> <lag-name> detail** command.

```
supervisor@rtbrick: op> show lag mode lcap lag-4
```

5. Show Commands

5.1. Instance Commands



The instance-name will be truncated in summary format after certain characters. In such cases full name can be found with detailed command output.

5.1.1. show instance

This command displays all configured instance information in summary format.

```
supervisor@rtbrick: op> show instance
Instance          Router ID      Instance ID   Status
default          192.1.0.3     0             Active
ip2vrf           192.1.4.3     2             Active
li-vrf           192.1.3.3     3             Active
mgmt-vrf         192.1.1.3     4             Active
radius-vrf       192.1.2.3     5             Active
supervisor@rtbrick: op>
```

5.1.2. show instance <instance-name>

This command displays the specified instance-name information in summary format.

```
supervisor@rtbrick: op> show instance ip2vrf
Instance          Router ID      Instance ID   Status
ip2vrf           192.1.4.3     2             Active
supervisor@rtbrick: op>
```

5.1.3. show instance detail

This command displays all configured instance information in detailed format.

```

supervisor@rtbrick: op> show instance detail
Instance: default
  Router ID: 192.1.0.3
  Instance ID: 0
  State: Active
  AFI      SAFI      State      Table ID
  ipv4     unicast   Active     0
  ipv4     multicast Active     0
  ipv4     labeled-unicast Active     1
  ipv6     unicast   Active     0
  ipv6     multicast Active     0
  ipv6     labeled-unicast Active     2
  mpls     unicast   Active     0
Instance: ip2vrf
  Router ID: 192.1.4.3
  Instance ID: 2
  State: Active
  AFI      SAFI      State      Table ID
  ipv4     unicast   Active     3
  ipv4     labeled-unicast Active     4
  ipv6     unicast   Active     5
  ipv6     labeled-unicast Active     6
  mpls     unicast   Active     7

```

5.1.4. show instance <instance-name> detail

This command displays the specified instance-name information in detailed format.

```

supervisor@rtbrick: op> show instance ip2vrf detail
Instance: ip2vrf
  Router ID: 192.1.4.3
  Instance ID: 2
  State: Active
  AFI      SAFI      State      Table ID
  ipv4     unicast   Active     3
  ipv4     labeled-unicast Active     4
  ipv6     unicast   Active     5
  ipv6     labeled-unicast Active     6
  mpls     unicast   Active     7

```

5.2. Neighbor Commands

5.2.1. show neighbor

This command displays the ipv4/ipv6 neighbor information in summary format.


```

supervisor@rtbrick: op> show neighbor
Instance           MAC Address           Interface           IP Address
Dynamic  Entry Time
default           7a:3f:c4:c2:03:01    if1-0/0/2/23
fe80::783f:c4ff:fec2:301  true    Wed Nov 11 09:31:54
default           b8:6a:97:a5:a0:35    if1-0/0/50/13
fe80::ba6a:97ff:fea5:a035  true    Wed Nov 11 09:31:57
default           b8:6a:97:a5:a0:39    if1-0/0/51/131
fe80::ba6a:97ff:fea5:a039  true    Wed Nov 11 09:32:03
ip2vrf           00:19:01:00:00:01    if1-0/0/4/1
192.168.231.1      true    Thu Nov 12 05:08:55

```

5.2.2. show neighbor <afi>

This command displays the specified address family neighbor information in summary format, where AFI can be ipv4 or ipv6.

```

supervisor@rtbrick: op> show neighbor ipv4
Instance           MAC Address           Interface           IP Address
Dynamic  Entry Time
ip2vrf           00:19:01:00:00:01    if1-0/0/4/1
192.168.231.1      true    Thu Nov 12 05:08:55
supervisor@rtbrick: op>

```

5.2.3. show neighbor <afi> instance <instance-name>

This command displays the specified address family (<afi>) and instance (<instance-name>) neighbor information in summary format, where AFI can be ipv4 or ipv6.

```

supervisor@rtbrick: op> show neighbor ipv6 instance default
Instance           MAC Address           Interface           IP Address
Dynamic  Entry Time
default           7a:3f:c4:c2:03:01    if1-0/0/2/23
fe80::783f:c4ff:fec2:301  true    Wed Nov 11 09:31:54
default           b8:6a:97:a5:a0:35    if1-0/0/50/13
fe80::ba6a:97ff:fea5:a035  true    Wed Nov 11 09:31:57
default           b8:6a:97:a5:a0:39    if1-0/0/51/131
fe80::ba6a:97ff:fea5:a039  true    Wed Nov 11 09:32:03
supervisor@rtbrick: op>

```

5.2.4. show neighbor instance <instance-name>

This command displays the specified instance-name neighbor information.

```

supervisor@rtbrick: op> show neighbor instance default
Instance           MAC Address           Interface           IP Address
Dynamic  Entry Time
default           7a:3f:c4:c2:03:01    if1-0/0/2/23
fe80::783f:c4ff:fec2:301  true    Wed Nov 11 09:31:54
default           b8:6a:97:a5:a0:35    if1-0/0/50/13
fe80::ba6a:97ff:fea5:a035  true    Wed Nov 11 09:31:57
default           b8:6a:97:a5:a0:39    if1-0/0/51/131
fe80::ba6a:97ff:fea5:a039  true    Wed Nov 11 09:32:03

```

5.3. Address Resolution Commands

5.3.1. show address-resolution request

This command displays address-resolution request table content in summary format.

```

supervisor@cli1: op> show address-resolution request
TableName: global.mribd.1.address.resolution.request
Next Hop           AFI           SAFI           Instance
100.1.1.1          ipv4          unicast        default
TableName: global.static.1.address.resolution.request
Next Hop           AFI           SAFI           Instance
10.1.1.1           ipv4          unicast        default
20.1.1.1           ipv4          unicast        default
30.1.1.1           ipv4          unicast        default
100.1.1.1          ipv4          unicast        default

```

5.3.2. show address-resolution response

This command displays address-resolution response table content in summary format.

```

supervisor@cli1: op> show address-resolution response
TableName: global.mribd.1.address.resolution.response
IP Address          Covering Prefix      MAC Address
Interface
100.1.1.1           100.1.1.1/32        7a:41:7e:60:00:08
memif-8/8/8/8
TableName: global.static.1.address.resolution.response
IP Address          Covering Prefix      MAC Address
Interface
10.1.1.1            10.1.1.1/32         7a:41:7e:61:01:01
ifl-0/0/52/1
20.1.1.1            20.1.1.1/32         7a:41:7e:62:02:02
ifl-0/0/53/1
30.1.1.1            30.1.1.1/32         7a:41:7e:63:03:03
ifl-0/0/54/1
100.1.1.1           100.1.1.1/32        7a:41:7e:60:00:08
ifl-0/0/55/1

```

5.4. Route Show Commands

5.4.1. show route

This command displays route from all address family sub address family and instance in summary format.

```

supervisor@rtbrick: op> show route
Instance: default, AFI: ipv4, SAFI: unicast
Prefix/Label          Source          Pref  Next Hop
Interface
192.1.1.0.3/32        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
192.1.1.0.3/32        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::1/128          bgp             20
fe80::ba6a:97ff:fea5:a035  ifl-0/0/50/13
192:1::2/128          bgp             20
fe80::783f:c4ff:fec2:301  ifl-0/0/2/23
192:1::3/128          direct          0     192:1::3
lo-0/0/0/1
192:1::4/128          bgp             20
fe80::ba6a:97ff:fea5:a035  ifl-0/0/50/13

```

5.4.2. show route detail

This command displays route from all address family sub address family and instance in detailed format.

```

supervisor@rtbrick: op> show route detail
Instance: default, AFI: ipv4, SAFI: unicast
192.1.0.3/32
  Source: direct, Preference: 0
  Next Hop: 192.1.0.3
    Next Hop type: glean, Next Hop action: trap to cpu
    Resolved in: default-ipv4-unicast
    Egress interface: lo-0/0/0/1

Instance: default, AFI: ipv4, SAFI: labeled-unicast
192.1.0.3/32
  Source: direct, Preference: 0
  Next Hop: 192.1.0.3
    Next Hop type: glean, Next Hop action: trap to cpu
    Resolved in: default-ipv4-labeled-unicast
    Egress interface: lo-0/0/0/1

Instance: default, AFI: ipv6, SAFI: unicast
192:1::1/128
  Source: bgp, Preference: 20
  Next Hop: fe80::ba6a:97ff:fea5:a035
    Covering prefix: fe80::ba6a:97ff:fea5:a035/128
    Next Hop type: direct, Next Hop action: None
    Resolved in: default-ipv6-unicast
    Egress interface: if1-0/0/50/13, NextHop MAC: b8:6a:97:a5:a0:35
    Created: Thu Nov 12 05:08:57 GMT +0000 2020

```

5.4.3. show route prefix <prefix/label>

This command displays the specified prefix information from all address family, sub address family and instance in summary format.

```

supervisor@rtbrick: op> show route prefix 192:1:2::3/128
Instance: radius-vrf, AFI: ipv6, SAFI: unicast
Prefix/Label          Source          Pref    Next Hop
Interface
192:1:2::3/128        direct         0       192:1:2::3
lo-0/0/2/1

Instance: radius-vrf, AFI: ipv6, SAFI: labeled-unicast
Prefix/Label          Source          Pref    Next Hop
Interface             Label
192:1:2::3/128        direct         0       192:1:2::3
lo-0/0/2/1           -

```

The 'show route prefix' commands provide a route lookup. If for a given prefix

there is no exact or longer match in the routing table, the commands will display the less-specific route entry that is used to forward the prefix.

In the following example, there is no route matching 192.168.1.0/30. This prefix will be forwarded using the less specific /24 route:

```
supervisor@rtbrick: op> show route ipv4 unicast instance services prefix
192.168.1.0/30
Instance: services, AFI: ipv4, SAFI: unicast
Prefix/Label                               Source                Pref    Next Hop
Interface
192.168.1.0/24                             bgp                   20
fd3d:3d:0:99::3                             memif-0/1/1/1
```

5.4.4. show route prefix <prefix/label> detail

This command displays the specified prefix information from all address family, sub address family and instance in detail format.

```
supervisor@rtbrick: op> show route prefix 192:1:2::3/128 detail
Instance: radius-vrf, AFI: ipv6, SAFI: unicast
192:1:2::3/128
  Source: direct, Preference: 0
  Next Hop: 192:1:2::3
    Next Hop type: glean, Next Hop action: trap to cpu
    Resolved in: radius-vrf-ipv6-unicast
    Egress interface: lo-0/0/2/1

Instance: radius-vrf, AFI: ipv6, SAFI: labeled-unicast
192:1:2::3/128
  Source: direct, Preference: 0
  Next Hop: 192:1:2::3
    Next Hop type: glean, Next Hop action: trap to cpu
    Resolved in: radius-vrf-ipv6-labeled-unicast
    Egress interface: lo-0/0/2/1
```

5.4.5. show route source <source>

This command displays routes from all address family, sub address family and instance with given route source in summary format.

```

supervisor@rtbrick: op> show route source bgp
Instance: default, AFI: ipv6, SAFI: unicast
Prefix/Label                               Source                               Pref   Next Hop
Interface
192:1::1/128                               bgp                                  20
fe80::ba6a:97ff:fea5:a035                 ifl-0/0/50/13
192:1::2/128                               bgp                                  20
fe80::783f:c4ff:fec2:301                 ifl-0/0/2/23
192:1::4/128                               bgp                                  20
fe80::ba6a:97ff:fea5:a035                 ifl-0/0/50/13

Instance: default, AFI: ipv6, SAFI: labeled-unicast
Prefix/Label                               Source                               Pref   Next Hop
Interface           Label
192:1::1/128       bgp                                  20
fe80::ba6a:97ff:fea5:a035   ifl-0/0/50/13   -
192:1::2/128       bgp                                  20
fe80::783f:c4ff:fec2:301   ifl-0/0/2/23   -
192:1::4/128       bgp                                  20
fe80::ba6a:97ff:fea5:a035   ifl-0/0/50/13   2004

```

5.4.6. show route source <source> detail

This command displays routes from all address family, sub address family and instance for the specified route source in detailed format.

```

supervisor@rtbrick: op> show route source bgp detail
Instance: default, AFI: ipv6, SAFI: unicast
192:1::1/128
  Source: bgp, Preference: 20
  Next Hop: fe80::ba6a:97ff:fea5:a035
  Covering prefix: fe80::ba6a:97ff:fea5:a035/128
  Next Hop type: direct, Next Hop action: None
  Resolved in: default-ipv6-unicast
  Egress interface: ifl-0/0/50/13, NextHop MAC: b8:6a:97:a5:a0:35
  Created: Thu Nov 12 05:08:57 GMT +0000 2020
192:1::2/128
  Source: bgp, Preference: 20
  Next Hop: fe80::783f:c4ff:fec2:301
  Covering prefix: fe80::783f:c4ff:fec2:301/128
  Next Hop type: direct, Next Hop action: None
  Resolved in: default-ipv6-unicast
  Egress interface: ifl-0/0/2/23, NextHop MAC: 7a:3f:c4:c2:03:01
  Created: Thu Nov 12 05:08:57 GMT +0000 2020

```

5.4.7. show route source <source> prefix <prefix>

This command displays route information for the specified source and prefix.

```

supervisor@rtbrick: op> show route source bgp prefix 192:1:2::1/128
Instance: radius-vrf, AFI: ipv6, SAFI: unicast
Prefix/Label          Source          Pref  Next Hop
Interface
192:1:2::1/128      bgp             20    192:1::1
ifl-0/0/50/13

```

5.4.8. show route source <source> prefix <prefix> detail

This command displays route information for the specified source and prefix in detailed format.

```

supervisor@rtbrick: op> show route source bgp prefix 192:1:2::1/128 detail
Instance: radius-vrf, AFI: ipv6, SAFI: unicast
192:1:2::1/128
  Source: bgp, Preference: 20
  Next Hop: 192:1::1
  Covering prefix: 192:1::1/128
  Next Hop type: mpls ingress, Next Hop action: mpls label push
  Resolved in: default-ipv6-labeled-unicast
  Egress interface: ifl-0/0/50/13, NextHop MAC: b8:6a:97:a5:a0:35
  MPLS-Label: 20023,bos:1
  Created: Thu Nov 12 05:08:57 GMT +0000 2020

```

5.4.9. show route source <source> instance <instance-name>

This command displays route information for the specified source and instance name.

```

supervisor@rtbrick: op> show route source bgp instance radius-vrf
Instance: radius-vrf, AFI: ipv4, SAFI: unicast
Prefix/Label          Source          Pref  Next Hop
Interface
0.0.0.0/0             bgp             20
fe80::ba6a:97ff:fea5:a035  ifl-0/0/50/13
192.1.2.1/32          bgp             20
fe80::ba6a:97ff:fea5:a035  ifl-0/0/50/13
192.1.2.2/32          bgp             20    192:1::2
ifl-0/0/2/23
192.1.2.4/32          bgp             20    192:1::4
ifl-0/0/50/13

```

5.4.10. show route source <source> instance <instance-name> detail

This command displays route information for the specified source, instance name in detailed format.

```

supervisor@rtbrick: op> show route source bgp instance radius-vrf detail
Instance: radius-vrf, AFI: ipv4, SAFI: unicast
0.0.0.0/0
  Source: bgp, Preference: 20
    Next Hop: fe80::ba6a:97ff:fea5:a035
      Covering prefix: fe80::ba6a:97ff:fea5:a035/128
      Next Hop type: mpls ingress, Next Hop action: mpls label push
      Resolved in: default-ipv6-labeled-unicast
      Egress interface: ifl-0/0/50/13, NextHop MAC: b8:6a:97:a5:a0:35
      MPLS-Label: 20021,bos:1
      Created: Thu Nov 12 05:08:57 GMT +0000 2020
192.1.2.1/32
  Source: bgp, Preference: 20
    Next Hop: fe80::ba6a:97ff:fea5:a035
      Covering prefix: fe80::ba6a:97ff:fea5:a035/128
      Next Hop type: mpls ingress, Next Hop action: mpls label push
      Resolved in: default-ipv6-labeled-unicast
      Egress interface: ifl-0/0/50/13, NextHop MAC: b8:6a:97:a5:a0:35
      MPLS-Label: 20021,bos:1
      Created: Thu Nov 12 05:08:57 GMT +0000 2020

```

5.4.11. show route source <source> instance <instance-name> prefix <prefix>

This command displays route information for the specified source, instance name and prefix.

```

supervisor@rtbrick: op> show route source bgp instance radius-vrf prefix
192.1.2.1/32
Instance: radius-vrf, AFI: ipv4, SAFI: unicast
Prefix/Label          Source          Pref    Next Hop
Interface
192.1.2.1/32          bgp             20
fe80::ba6a:97ff:fea5:a035  ifl-0/0/50/13

```

5.4.12. show route source <source> instance <instance-name> prefix <prefix> detail

This command displays route information for the specified source, instance name and prefix in detailed format.


```

supervisor@rtbrick: op> show route source bgp instance radius-vrf prefix
192.1.2.1/32 detail
Instance: radius-vrf, AFI: ipv4, SAFI: unicast
192.1.2.1/32
  Source: bgp, Preference: 20
    Next Hop: fe80::ba6a:97ff:fea5:a035
      Covering prefix: fe80::ba6a:97ff:fea5:a035/128
      Next Hop type: mpls ingress, Next Hop action: mpls label push
      Resolved in: default-ipv6-labeled-unicast
      Egress interface: if1-0/0/50/13, NextHop MAC: b8:6a:97:a5:a0:35
      MPLS-Label: 20021,bos:1
      Created: Thu Nov 12 05:08:57 GMT +0000 2020

```

5.4.13. show route instance <instance-name>

This command displays route information for the specified instance name.

```

supervisor@rtbrick: op> show route instance li-vrf
Instance: li-vrf, AFI: ipv4, SAFI: unicast
Prefix/Label                Source                Pref    Next Hop
Interface
0.0.0.0/0                   bgp                   20
fe80::ba6a:97ff:fea5:a035   if1-0/0/50/13
192.1.3.1/32                bgp                   20
fe80::ba6a:97ff:fea5:a035   if1-0/0/50/13
192.1.3.2/32                bgp                   20      192:1::2
if1-0/0/2/23
192.1.3.3/32                direct                0       192.1.3.3
lo-0/0/3/1
192.1.3.4/32                bgp                   20      192:1::4
if1-0/0/50/13

```

5.4.14. show route instance <instance-name> detail

This command displays route information for the specified instance name in detailed format.

```

supervisor@rtbrick: op> show route instance li-vrf detail
Instance: li-vrf, AFI: ipv4, SAFI: unicast
0.0.0.0/0
  Source: bgp, Preference: 20
  Next Hop: fe80::ba6a:97ff:fea5:a035
    Covering prefix: fe80::ba6a:97ff:fea5:a035/128
    Next Hop type: mpls ingress, Next Hop action: mpls label push
    Resolved in: default-ipv6-labeled-unicast
    Egress interface: ifl-0/0/50/13, NextHop MAC: b8:6a:97:a5:a0:35
    MPLS-Label: 20009,bos:1
    Created: Thu Nov 12 05:08:57 GMT +0000 2020
192.1.3.1/32
  Source: bgp, Preference: 20
  Next Hop: fe80::ba6a:97ff:fea5:a035
    Covering prefix: fe80::ba6a:97ff:fea5:a035/128
    Next Hop type: mpls ingress, Next Hop action: mpls label push
    Resolved in: default-ipv6-labeled-unicast
    Egress interface: ifl-0/0/50/13, NextHop MAC: b8:6a:97:a5:a0:35
    MPLS-Label: 20009,bos:1
    Created: Thu Nov 12 05:08:57 GMT +0000 2020

```

5.4.15. show route instance <instance-name> prefix <prefix>

This command displays route information for the specified instance name and prefix.

```

supervisor@rtbrick: op> show route instance li-vrf prefix 192:1:3::3/128
Instance: li-vrf, AFI: ipv6, SAFI: unicast
Prefix/Label          Source          Pref    Next Hop
Interface
192:1:3::3/128        direct          0       192:1:3::3
lo-0/0/3/1

Instance: li-vrf, AFI: ipv6, SAFI: labeled-unicast
Prefix/Label          Source          Pref    Next Hop
Interface             Label
192:1:3::3/128        direct          0       192:1:3::3
lo-0/0/3/1            -

```

5.4.16. show route instance <instance-name> prefix <prefix> detail

This command displays route information for the specified instance name and prefix in detailed format.

```

supervisor@rtbrick: op> show route instance li-vrf prefix 192:1:3::3/128
detail
Instance: li-vrf, AFI: ipv6, SAFI: unicast
192:1:3::3/128
  Source: direct, Preference: 0
  Next Hop: 192:1:3::3
    Next Hop type: glean, Next Hop action: trap to cpu
    Resolved in: li-vrf-ipv6-unicast
    Egress interface: lo-0/0/3/1

Instance: li-vrf, AFI: ipv6, SAFI: labeled-unicast
192:1:3::3/128
  Source: direct, Preference: 0
  Next Hop: 192:1:3::3
    Next Hop type: glean, Next Hop action: trap to cpu
    Resolved in: li-vrf-ipv6-labeled-unicast
    Egress interface: lo-0/0/3/1

```

5.4.17. show route <afi>

This command displays route information for the specified address family identifier.

```

supervisor@rtbrick: op> show route ipv4
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            -

```

5.4.18. show route <afi> detail

This command displays route information for the specified address family identifier in detailed format.

```

supervisor@rtbrick: op> show route ipv4 detail
Instance: default, AFI: ipv4, SAFI: unicast
192.1.0.3/32
  Source: direct, Preference: 0
  Next Hop: 192.1.0.3
    Next Hop type: glean, Next Hop action: trap to cpu
    Resolved in: default-ipv4-unicast
    Egress interface: lo-0/0/0/1

Instance: default, AFI: ipv4, SAFI: labeled-unicast
192.1.0.3/32
  Source: direct, Preference: 0
  Next Hop: 192.1.0.3
    Next Hop type: glean, Next Hop action: trap to cpu
    Resolved in: default-ipv4-labeled-unicast
    Egress interface: lo-0/0/0/1

```

5.4.19. show route <afi> prefix <prefix>

This command displays route information for the specified address family identifier and prefix.

```

supervisor@rtbrick: op> show route ipv4 prefix 192.1.2.3/32
Instance: radius-vrf, AFI: ipv4, SAFI: unicast
Prefix/Label          Source          Pref  Next Hop
Interface
192.1.2.3/32         direct         0    192.1.2.3
lo-0/0/2/1

Instance: radius-vrf, AFI: ipv4, SAFI: labeled-unicast
Prefix/Label          Source          Pref  Next Hop
Interface             Label
192.1.2.3/32         direct         0    192.1.2.3
lo-0/0/2/1           -

```

5.4.20. show route <afi> prefix <prefix> detail

This command displays route information for the specified address family identifier and prefix in detailed format.

```

supervisor@rtbrick: op> show route ipv4 prefix 192.1.2.3/32 detail
Instance: radius-vrf, AFI: ipv4, SAFI: unicast
192.1.2.3/32
  Source: direct, Preference: 0
    Next Hop: 192.1.2.3
      Next Hop type: glean, Next Hop action: trap to cpu
      Resolved in: radius-vrf-ipv4-unicast
      Egress interface: lo-0/0/2/1

Instance: radius-vrf, AFI: ipv4, SAFI: labeled-unicast
192.1.2.3/32
  Source: direct, Preference: 0
    Next Hop: 192.1.2.3
      Next Hop type: glean, Next Hop action: trap to cpu
      Resolved in: radius-vrf-ipv4-labeled-unicast
      Egress interface: lo-0/0/2/1
    
```

5.4.21. show route <afi> source <source>

This command displays route information for the specified address family identifier and source.

```

supervisor@rtbrick: op> show route ipv4 source direct
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            -
    
```

5.4.22. show route <afi> source <source> detail

This command displays route information for the specified address family identifier and source in detailed format.

```

supervisor@rtbrick: op> show route ipv4 source direct detail
Instance: default, AFI: ipv4, SAFI: unicast
192.1.0.3/32
  Source: direct, Preference: 0
  Next Hop: 192.1.0.3
    Next Hop type: glean, Next Hop action: trap to cpu
    Resolved in: default-ipv4-unicast
    Egress interface: lo-0/0/0/1

Instance: default, AFI: ipv4, SAFI: labeled-unicast
192.1.0.3/32
  Source: direct, Preference: 0
  Next Hop: 192.1.0.3
    Next Hop type: glean, Next Hop action: trap to cpu
    Resolved in: default-ipv4-labeled-unicast
    Egress interface: lo-0/0/0/1

```

5.4.23. show route <afi> source <source> prefix <prefix>

This command displays route information for the specified address family identifier, source, and prefix.

```

supervisor@rtbrick: op> show route ipv4 source direct prefix 192.1.2.3/32
Instance: radius-vrf, AFI: ipv4, SAFI: unicast
Prefix/Label          Source          Pref    Next Hop
Interface
192.1.2.3/32         direct          0       192.1.2.3
lo-0/0/2/1

Instance: radius-vrf, AFI: ipv4, SAFI: labeled-unicast
Prefix/Label          Source          Pref    Next Hop
Interface             Label
192.1.2.3/32         direct          0       192.1.2.3
lo-0/0/2/1           -

```

5.4.24. show route <afi> source <source> prefix <prefix> detail

This command displays route information for the specified address family identifier, source, and prefix in detailed format.

```

supervisor@rtbrick: op> show route ipv4 source direct prefix 192.1.2.3/32
detail
Instance: radius-vrf, AFI: ipv4, SAFI: unicast
192.1.2.3/32
  Source: direct, Preference: 0
  Next Hop: 192.1.2.3
  Next Hop type: glean, Next Hop action: trap to cpu
  Resolved in: radius-vrf-ipv4-unicast
  Egress interface: lo-0/0/2/1

Instance: radius-vrf, AFI: ipv4, SAFI: labeled-unicast
192.1.2.3/32
  Source: direct, Preference: 0
  Next Hop: 192.1.2.3
  Next Hop type: glean, Next Hop action: trap to cpu
  Resolved in: radius-vrf-ipv4-labeled-unicast
  Egress interface: lo-0/0/2/1

```

5.4.25. show route <afi> source <source> instance <instance-name>

This command displays route information for the specified address family identifier, source, and instance name.

```

supervisor@rtbrick: op> show route ipv4 source direct instance mgmt-vrf
Instance: mgmt-vrf, AFI: ipv4, SAFI: unicast
Prefix/Label          Source          Pref    Next Hop
Interface
192.1.1.3/32          direct          0       192.1.1.3
lo-0/0/1/1

Instance: mgmt-vrf, AFI: ipv4, SAFI: labeled-unicast
Prefix/Label          Source          Pref    Next Hop
Interface             Label
192.1.1.3/32          direct          0       192.1.1.3
lo-0/0/1/1           -
supervisor@rtbrick: op>

```

5.4.26. show route <afi> source <source> instance <instance-name> detail

This command displays route information for the specified address family identifier, source, and instance name in detailed format.

```

supervisor@rtbrick: op> show route ipv6 source direct instance mgmt-vrf
detail
Instance: mgmt-vrf, AFI: ipv6, SAFI: unicast
192:1:1::3/128
  Source: direct, Preference: 0
  Next Hop: 192:1:1::3
  Next Hop type: glean, Next Hop action: trap to cpu
  Resolved in: mgmt-vrf-ipv6-unicast
  Egress interface: lo-0/0/1/1

Instance: mgmt-vrf, AFI: ipv6, SAFI: labeled-unicast
192:1:1::3/128
  Source: direct, Preference: 0
  Next Hop: 192:1:1::3
  Next Hop type: glean, Next Hop action: trap to cpu
  Resolved in: mgmt-vrf-ipv6-labeled-unicast
  Egress interface: lo-0/0/1/1

```

5.4.27. show route <afi> source <source> instance <instance-name> prefix <prefix>

This command displays route information for the specified address family identifier, instance and prefix.

```

supervisor@rtbrick: op> show route ipv6 source direct instance mgmt-vrf
prefix 192:1:1::3/128
Instance: mgmt-vrf, AFI: ipv6, SAFI: unicast
Prefix/Label          Source          Pref    Next Hop
Interface
192:1:1::3/128       direct          0       192:1:1::3
lo-0/0/1/1

Instance: mgmt-vrf, AFI: ipv6, SAFI: labeled-unicast
Prefix/Label          Source          Pref    Next Hop
Interface             Label
192:1:1::3/128       direct          0       192:1:1::3
lo-0/0/1/1           -

```

5.4.28. show route <afi> source <source> instance <instance-name> prefix <prefix> detail

This command displays route information for the specified address family identifier, instance and prefix in detailed format.


```

supervisor@rtbrick: op> show route ipv6 source direct instance mgmt-vrf
prefix 192:1:1::3/128 detail
Instance: mgmt-vrf, AFI: ipv6, SAFI: unicast
192:1:1::3/128
  Source: direct, Preference: 0
  Next Hop: 192:1:1::3
  Next Hop type: glean, Next Hop action: trap to cpu
  Resolved in: mgmt-vrf-ipv6-unicast
  Egress interface: lo-0/0/1/1

Instance: mgmt-vrf, AFI: ipv6, SAFI: labeled-unicast
192:1:1::3/128
  Source: direct, Preference: 0
  Next Hop: 192:1:1::3
  Next Hop type: glean, Next Hop action: trap to cpu
  Resolved in: mgmt-vrf-ipv6-labeled-unicast
  Egress interface: lo-0/0/1/1

```

5.4.29. show route <afi> <safi>

This command displays route information for the specified address family.

```

supervisor@rtbrick: op> show route mpls unicast
Instance: default, AFI: mpls, SAFI: unicast
Prefix/Label          Source          Pref   Next Hop
Interface
2001                   bgp             170
fe80::ba6a:97ff:fea5:a035  if1-0/0/50/13
2002                   bgp             170
fe80::783f:c4ff:fec2:301  if1-0/0/2/23
2004                   bgp             170
fe80::ba6a:97ff:fea5:a035  if1-0/0/50/13

```

5.4.30. show route <afi> <safi> detail

This command displays route information for the specified address family in detailed format.

```

supervisor@rtbrick: op> show route mpls unicast detail
Instance: default, AFI: mpls, SAFI: unicast
2001
  Source: bgp, Preference: 170
  Next Hop: fe80::ba6a:97ff:fea5:a035
  Covering prefix: fe80::ba6a:97ff:fea5:a035/128
  Next Hop type: mpls php, Next Hop action: mpls label pop - next
protocol ipv6
  Resolved in: default-ipv6-labeled-unicast
  Egress interface: if1-0/0/50/13, NextHop MAC: b8:6a:97:a5:a0:35
  Created: Thu Nov 12 05:08:57 GMT +0000 2020
2002
  Source: bgp, Preference: 170
  Next Hop: fe80::783f:c4ff:fec2:301
  Covering prefix: fe80::783f:c4ff:fec2:301/128
  Next Hop type: mpls php, Next Hop action: mpls label pop - next
protocol ipv6
  Resolved in: default-ipv6-labeled-unicast
  Egress interface: if1-0/0/2/23, NextHop MAC: 7a:3f:c4:c2:03:01
  Created: Thu Nov 12 05:08:57 GMT +0000 2020

```

5.4.31. show route <afi> <safi> prefix <prefix>

This command displays route information for the specified address family and prefix.

```

supervisor@rtbrick: op> show route ipv6 unicast prefix 192:1:4::3/128
Instance: ip2vrf, AFI: ipv6, SAFI: unicast
Prefix/Label          Source          Pref    Next Hop
Interface
192:1:4::3/128       direct         0       192:1:4::3
lo-0/0/4/1

32) show route <afi> <safi> prefix <prefix> detail
supervisor@rtbrick: op> show route ipv6 unicast prefix 192:1:4::3/128 detail

Instance: ip2vrf, AFI: ipv6, SAFI: unicast
192:1:4::3/128
  Source: direct, Preference: 0
  Next Hop: 192:1:4::3
  Next Hop type: glean, Next Hop action: trap to cpu
  Resolved in: ip2vrf-ipv6-unicast
  Egress interface: lo-0/0/4/1

```

5.4.32. show route <afi> <safi> source <source>

This command displays route information for the specified address family and source.

```

supervisor@rtbrick: op> show route ipv6 unicast source direct
Instance: default, AFI: ipv6, SAFI: unicast
Prefix/Label          Source          Pref  Next Hop
Interface
192:1::3/128          direct          0     192:1::3
lo-0/0/0/1

Instance: ip2vrf, AFI: ipv6, SAFI: unicast
Prefix/Label          Source          Pref  Next Hop
Interface
192:1:4::3/128        direct          0     192:1:4::3
lo-0/0/4/1

```

5.4.33. show route <afi> <safi> source <source> detail

This command displays route information for the specified address family, and source in detailed format.

```

supervisor@rtbrick: op> show route ipv6 unicast source direct detail
Instance: default, AFI: ipv6, SAFI: unicast
192:1::3/128
  Source: direct, Preference: 0
  Next Hop: 192:1::3
    Next Hop type: glean, Next Hop action: trap to cpu
    Resolved in: default-ipv6-unicast
    Egress interface: lo-0/0/0/1

```

5.4.34. show route <afi> <safi> source <source> prefix <prefix>

This command displays route information for the specified address family, source, and prefix.

```

supervisor@rtbrick: op> show route ipv6 unicast source direct prefix
192:1:4::3/128
Instance: ip2vrf, AFI: ipv6, SAFI: unicast
Prefix/Label          Source          Pref  Next Hop
Interface
192:1:4::3/128        direct          0     192:1:4::3
lo-0/0/4/1

```

5.4.35. show route <afi> <safi> source <source> prefix <prefix> detail

This command displays route information for the specified address family, source, and prefix in detailed format.

```

supervisor@rtbrick: op> show route ipv6 unicast source direct prefix
192:1:4::3/128 detail
Instance: ip2vrf, AFI: ipv6, SAFI: unicast
192:1:4::3/128
  Source: direct, Preference: 0
    Next Hop: 192:1:4::3
      Next Hop type: glean, Next Hop action: trap to cpu
      Resolved in: ip2vrf-ipv6-unicast
      Egress interface: lo-0/0/4/1

```

5.4.36. show route <afi> <safi> source <source> instance <instance-name>

This command displays route information for the specified address family, source, and instance name.

```

supervisor@rtbrick: op> show route ipv6 unicast source direct instance ip2vrf
Instance: ip2vrf, AFI: ipv6, SAFI: unicast
Prefix/Label                               Source           Pref    Next Hop
Interface
192:1:4::3/128                             direct           0       192:1:4::3
lo-0/0/4/1

```

5.4.37. show route <afi> <safi> source <source> instance <instance-name> detail

This command displays route information for the specified address family, source, and instance name in detailed format.

```

supervisor@rtbrick: op> show route ipv6 unicast source direct instance ip2vrf
detail
Instance: ip2vrf, AFI: ipv6, SAFI: unicast
192:1:4::3/128
  Source: direct, Preference: 0
    Next Hop: 192:1:4::3
      Next Hop type: glean, Next Hop action: trap to cpu
      Resolved in: ip2vrf-ipv6-unicast
      Egress interface: lo-0/0/4/1

```

5.4.38. show route <afi> <safi> source <source> instance <instance-name> prefix <prefix>

This command displays route information for the specified address family, source, instance, and prefix.

```
supervisor@rtbrick: op> show route ipv6 unicast source direct instance ip2vrf
prefix 192:1:4::3/128
Instance: ip2vrf, AFI: ipv6, SAFI: unicast
Prefix/Label          Source          Pref    Next Hop
Interface
192:1:4::3/128        direct          0       192:1:4::3
lo-0/0/4/1
```

5.4.39. show route <afi> <safi> source <source> instance <instance-name> prefix <prefix> detail

This command displays route information for the specified address family, source, instance, and prefix in detailed format.

```
supervisor@rtbrick: op> show route ipv6 unicast source direct instance ip2vrf
prefix 192:1:4::3/128 detail
Instance: ip2vrf, AFI: ipv6, SAFI: unicast
192:1:4::3/128
  Source: direct, Preference: 0
  Next Hop: 192:1:4::3
    Next Hop type: glean, Next Hop action: trap to cpu
    Resolved in: ip2vrf-ipv6-unicast
    Egress interface: lo-0/0/4/1
```

6. ACL Show Commands

6.1. show acl

This command displays brief information about access-control list (ACL).

```
supervisor@rtbrick: op> show acl
ACL                               Ordinal   Type      Attach Point
lldp.ifp-0/0/4.trap.rule         -         12       ifp-0/0/4
lldp.ifp-0/0/17.trap.rule        -         12       ifp-0/0/17
lldp.ifp-0/0/53.trap.rule        -         12       ifp-0/0/53
```

6.2. show acl detail

This command displays detailed information about access-control list (ACL).

```
supervisor@rtbrick: op> show acl detail
Rule: lldp.ifp-0/0/4.trap.rule
  ACL type: 12
  Ordinal: -
  Match:
    Attachment point: ifp-0/0/4
    Direction: ingress
    Destination MAC: 01:80:c2:00:00:0e
  Action:
    Redirect to CPU: True
    Forward class: class-1
  Result:
    Trap ID: LLDP
Rule: lldp.ifp-0/0/17.trap.rule
  ACL type: 12
  Ordinal: -
  Match:
    Attachment point: ifp-0/0/17
    Direction: ingress
    Destination MAC: 01:80:c2:00:00:0e
  Action:
    Redirect to CPU: True
    Forward class: class-1
  Result:
    Trap ID: LLDP
Rule: lldp.ifp-0/0/53.trap.rule
  ACL type: 12
  Ordinal: -
  Match:
    Attachment point: ifp-0/0/53
    Direction: ingress
    Destination MAC: 01:80:c2:00:00:0e
  Action:
    Redirect to CPU: True
    Forward class: class-1
  Result:
    Trap ID: LLDP
```

6.3. show acl <acl-rule-name>

This command displays detailed information for a specified acl-rule-name.

```
supervisor@rtbrick: op> show acl lldp.ifp-0/0/53.trap.rule
Rule: lldp.ifp-0/0/53.trap.rule
  ACL type: 12
  Ordinal: -
  Match:
    Attachment point: ifp-0/0/53
    Direction: ingress
    Destination MAC: 01:80:c2:00:00:0e
  Action:
    Redirect to CPU: True
    Forward class: class-1
  Result:
    Trap ID: LLDP
supervisor@rtbrick: op>
```


7. ACL Statistics Commands

The ACL Statistics Commands are supported on Edgecore AS5916-54XKS platform only.

7.1. show acl statistics

This command displays ACL statistics information.

```
supervisor@rtbrick: op> show acl statistics
ACL                               Units      Total      Received   Dropped
lldp.ifp-0/0/1.trap.rule         Packets    -          -          -
                                   Bytes      -          -          -
lldp.ifp-0/0/2.trap.rule         Packets    -          -          -
                                   Bytes      -          -          -
lldp.ifp-0/0/3.trap.rule         Packets    -          -          -
                                   Bytes      -          -          -
lldp.ifp-0/0/4.trap.rule         Packets    -          -          -
                                   Bytes      -          -          -
lldp.ifp-0/0/40.trap.rule        Packets    -          -          -
                                   Bytes      -          -          -
lldp.ifp-0/0/50.trap.rule        Packets    -          -          -
                                   Bytes      -          -          -
lldp.ifp-0/0/51.trap.rule        Packets    -          -          -
                                   Bytes      -          -          -
lldp.ifp-0/0/52.trap.rule        Packets    -          -          -
                                   Bytes      -          -          -
lldp.ifp-0/0/53.trap.rule        Packets    -          -          -
                                   Bytes      -          -          -
lldp.ifp-0/0/54.trap.rule        Packets    -          -          -
                                   Bytes      -          -          -
```

7.2. show acl <acl-name> statistics

This command displays ACL statistics information for the specified ACL.

```
supervisor@rtbrick: op> show acl lldp.ifp-0/0/1.trap.rule statistics
ACL                               Units      Total      Received   Dropped
lldp.ifp-0/0/1.trap.rule         Packets    -          -          -
                                   Bytes      -          -          -
supervisor@rtbrick: op>
```

8. Interface Show Commands

8.1. show interface physical

This command displays all the physical interface including loopback, cpu and recycle ports.

```

supervisor@rtbrick: 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

```

8.2. show interface <interface-name>

This command displays summary output of interface and its logical interface if any.

```

supervisor@rtbrick: op> show interface ifp-0/0/52
Interface Admin Link Oper IPv4 Address IPv6 Address
ifp-0/0/52 Up Up Up
ifl-0/0/52/4 Up Up Up 4.4.4.4/24 fe80::82a2:35ff:feee:a83d/128

```

8.3. show interface logical

This command display all the logical interface in the system including loopback, cpu and recycle logical ports.

```

supervisor@rtbrick: 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

```

8.4. show interface logical instance default

This command displays all logical interface in the instance including loopback, cpu and recycle logical ports.

```

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
    
```

8.5. show interface address

This command displays all ipv4 and ipv6 address in the system.

```

supervisor@rtbrick: op> show interface address
Interface          Instance          IPv4 Address          IPv4
Primary IPv6 Address
ifl-0/0/10/100    default          2.2.2.2/24           True
fe80::82a2:35ff:feee:a80a/128
ifl-0/0/10/200    default          3.3.3.2/24           True
fe80::c9:80a2:35ff:feee:a80a/128
ifl-0/0/10/300    default          -
fe80::12d:80a2:35ff:feee:a80a/128
    
```

8.6. show interface address instance default

This command displays all address in the instance includes both ipv4 and ipv6.

```

supervisor@rtbrick: op> show interface address instance default
Interface          Instance          IPv4 Address          IPv4
Primary IPv6 Address
ifl-0/0/10/100    default          2.2.2.2/24           True
fe80::82a2:35ff:feee:a80a/128
ifl-0/0/10/200    default          3.3.3.2/24           True
fe80::c9:80a2:35ff:feee:a80a/128
ifl-0/0/10/300    default          -
fe80::12d:80a2:35ff:feee:a80a/128
    
```

8.7. show interface summary

This command displays output for all the interface including physical and logical and its configured address.

```

supervisor@rtbrick: 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	2.2.2.2/24
fe80::82a2:35ff:feee:a80a/128				
ifl-0/0/10/200	Up	Up	Up	3.3.3.2/24
fe80::c9:80a2:35ff:feee:a80a/128				
ifl-0/0/10/300	Up	Up	Up	-
fe80::12d:80a2:35ff:feee:a80a/128				
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
```

8.8. show interface <interface-name>

This command displays summarised output for specific physical port.

```
supervisor@rtbrick: 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       2.2.2.2/24
fe80::82a2:35ff:feee:a80a/128
  ifl-0/0/10/200    Up       Up       Up       3.3.3.2/24
fe80::c9:80a2:35ff:feee:a80a/128
  ifl-0/0/10/300    Up       Up       Up       -
fe80::12d:80a2:35ff:feee:a80a/128
  ifl-0/0/10/1000   Up       Up       Up       -
fe80::3e9:80a2:35ff:feee:a80a/128
```

8.9. show interface <interface-name> detail

This command displays detailed output for specific physical port and logical ports under it.

```

supervisor@rtbrick: op> show interface ifp-0/0/10 detail
Interface:ifp-0/0/52
Admin/Link/Operational status: Up/Up/Up
Speed configured: 100G
Speed maximum: 100G
Duplex: Full
Autonegotiation: Disabled
Encapsulation mode: ieee
MTU: 16360
Maximum frame size: 16360
Interface type: ethernet
Interface index: 124929
MAC: 80:a2:35:ee:a8:3d
Uptime: Tue Nov 17 14:10:46 GMT +0000 2020
Description: Physical interface #52 from node 0, chip 0
Packet statistics:
Rx packets: 16034 Tx packets: 17295
Rx bytes: 1602124 Tx bytes: 1711264
Interface:ifl-0/0/52/4, 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: 106497
MAC: 80:a2:35:ee:a8:3d

IPv4 Address IPv6 Address
4.4.4.4/24 fe80::82a2:35ff:feee:a83d/128

Packet statistics:
Ingress forwarded packets: 16000
Ingress forwarded bytes: 1598656
Ingress drop Packets: 0
Ingress drop bytes: 0
Egress forwarded packets: 0
Egress forwarded bytes: 0
Egress drop packets: 0
Egress drop bytes: 0

```

8.10. show interface <interface-name> statistics

This command displays detailed statistics output for specific physical port and logical ports under it.

```

supervisor@rtbrick: 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	-	-	Packets	-	-	-
-	Unicast Statistics	-	Bytes	-	-	-
-	-	-	Packets	-	-	-
-	Broadcast Statistics	-	Bytes	-	-	-
-	-	-	Packets	-	-	-
-	Multicast Statistics	-	Bytes	-	-	-
-	-	-	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
bcmQmxIfHCOutBroadcastPkts: 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	-	-	Packets	-	-	-
-	Unicast Statistics	-	Bytes	-	-	-
-	-	-	Packets	-	-	-
-	Broadcast Statistics	-	Bytes	-	-	-
-	-	-	Packets	-	-	-
-	Multicast Statistics	-	Bytes	-	-	-
-	-	-	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
supervisor@rtbrick: op>

```

8.11. show interface <interface-name> statistics

This command displays detailed statistics output for specific logical ports.

```

supervisor@rtbrick: op> show interface ifl-0/0/10/100 statistics
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 4998 - - -
- - - - - Bytes - - -
- Error - - RX Packets - - -
- - - - - Bytes - - -
- Error - - TX Packets 47 - - -
- - - - - Bytes - - -
- No Buff - - RX Packets - - -
- - - - - Bytes - - -
- Traffic Statistics - - Packets 4998 - - -
68522 - - - Bytes 489804 - - -
5872472 - - -
- Unicast Statistics - - Packets - - -
- - - - - Bytes - - -
- Broadcast Statistics - - Packets - - -
- - - - - Bytes - - -
- Multicast Statistics - - Packets - - -
- - - - - Bytes - - -
-
- Packet Statistics:
-   Ingress Forwarded Packets: 1813
-   Ingress Forwarded Bytes: 184926
-   Ingress Drop Packets: 1
-   Ingress Drop Bytes: 102
-   Egress Forwarded Packets: 0
-   Egress Forwarded Bytes: 0
-   Egress Drop Packets: 0
-   Egress Drop Bytes: 0
- supervisor@rtbrick: op>
- supervisor@rtbrick: op> show interface ifl-0/0/10/100 detail
- Interface:ifl-0/0/10/100, Instance:default
-   Admin/Link/Operational Status: Up/Up/Up
-   IPv4/IPv6/MPLS Status: Up/Up/Up
-   IPv4/IPv6/MPLS MTU: 1500/1500/1500
-   Interface Type: Logical Sub interface
-   Interface index: 20481
-   MAC: 80:a2:35:ee:a8:0a

```

```

Class-of-Service: pta_8queues_comp_on_S

IPv4 Address          IPv6 Address
2.2.2.2/24           fe80::82a2:35ff:feee:a80a/128

Packet Statistics:
  Ingress Forwarded Packets: 1813
  Ingress Forwarded Bytes: 184926
  Ingress Drop Packets: 1
  Ingress Drop Bytes: 102
  Egress Forwarded Packets: 0
  Egress Forwarded Bytes: 0
  Egress Drop Packets: 0
  Egress Drop Bytes: 0
supervisor@rtbrick: op>

```

8.12. show interface detail

This command displays the detailed output for all ports.

```

supervisor@rtbrick: op> show interface detail
Interface          Admin   Link   Oper          IPv4 Address
IPv6 Address
Interface:ifp-0/0/1
  Admin/Link/Operational Status: Up/Down/Down
  Speed configured: 10G
  Speed maximum: 10G
  Duplex: Full
  Autonegotiation: Disabled
  Encapsulation mode: ieee
  MTU: 16360
  Maximum Frame Size: 16360
  Interface Type: ethernet
  Interface index: 2049
  MAC: 80:a2:35:ee:a8:01
  Uptime: Wed Nov 11 14:55:53 GMT +0000 2020
  Description: Physical interface #1 from node 0, chip 0
  Packet Statistics:
    Rx Packets: 0          Tx Packets: 0
    Rx Bytes: 0           Tx Bytes: 0
Interface:ifp-0/0/2
  Admin/Link/Operational Status: Up/Down/Down
  Speed configured: 10G
  Speed maximum: 10G
  Duplex: Full
  Autonegotiation: Disabled
  Encapsulation mode: ieee
  MTU: 16360
  Maximum Frame Size: 16360
  Interface Type: ethernet
  Interface index: 4097
  MAC: 80:a2:35:ee:a8:02
  Uptime: Wed Nov 11 14:55:53 GMT +0000 2020
  Description: Physical interface #2 from node 0, chip 0
  Packet Statistics:

```



```

Rx Packets: 0          Tx Packets: 0
Rx Bytes: 0           Tx Bytes: 0
Interface:ifp-0/0/3
  Admin/Link/Operational Status: Up/Down/Down
  Speed configured: 10G
  Speed maximum: 10G
  Duplex: Full
  Autonegotiation: Disabled
  Encapsulation mode: ieee
  MTU: 16360
  Maximum Frame Size: 16360
  Interface Type: ethernet
  Interface index: 6145
  MAC: 80:a2:35:ee:a8:03
  Uptime: Wed Nov 11 14:55:53 GMT +0000 2020
  Description: Physical interface #3 from node 0, chip 0
  Packet Statistics:
    Rx Packets: 0          Tx Packets: 0
    Rx Bytes: 0           Tx Bytes: 0
Interface:ifp-0/0/4
  Admin/Link/Operational Status: Up/Up/Up
  Speed configured: 10G
  Speed maximum: 10G
  Duplex: Full
  Autonegotiation: Disabled
  Encapsulation mode: ieee
  MTU: 16360
  Maximum Frame Size: 16360
  Interface Type: ethernet
  Interface index: 8193
  MAC: 80:a2:35:ee:a8:04
  Uptime: Fri Nov 13 15:04:20 GMT +0000 2020
  Description: Physical interface #4 from node 0, chip 0
  Packet Statistics:
    Rx Packets: 5744      Tx Packets: 5748
    Rx Bytes: 574400     Tx Bytes: 574800
Interface:ifp-0/0/5
  Admin/Link/Operational Status: Up/Down/Down
  Speed configured: 10G
  Speed maximum: 10G
  Duplex: Full
  Autonegotiation: Disabled
  Encapsulation mode: ieee
  MTU: 16360
  Maximum Frame Size: 16360
  Interface Type: ethernet
  Interface index: 10241
  MAC: 80:a2:35:ee:a8:05
  Uptime: Wed Nov 11 14:55:53 GMT +0000 2020
  Description: Physical interface #5 from node 0, chip 0
  Packet Statistics:
    Rx Packets: 0          Tx Packets: 0
    Rx Bytes: 0           Tx Bytes: 0
Interface:ifp-0/0/6
  Admin/Link/Operational Status: Up/Down/Down
  Speed configured: 10G
  Speed maximum: 10G
  Duplex: Full

```

```

Autonegotiation: Disabled
Encapsulation mode: ieee
MTU: 16360
Maximum Frame Size: 16360
Interface Type: ethernet
Interface index: 12289
MAC: 80:a2:35:ee:a8:06
Uptime: Wed Nov 11 14:55:53 GMT +0000 2020
Description: Physical interface #6 from node 0, chip 0
Packet Statistics:
  Rx Packets: 0          Tx Packets: 0
  Rx Bytes: 0           Tx Bytes: 0

```

8.13. show interface statistics

This command displays summarized statistics output for all ports.

```

supervisor@rtbrick: op> show interface statistics
Interface:  ifp-0/0/1

```

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	-	-	-
-	-	-	Bytes	-	-	-
-	Unicast Statistics	-	Packets	-	-	-
-	-	-	Bytes	-	-	-
-	Broadcast Statistics	-	Packets	-	-	-
-	-	-	Bytes	-	-	-
-	Multicast Statistics	-	Packets	-	-	-
-	-	-	Bytes	-	-	-
-	Traffic Statistics	-	Packets	-	-	-
-	-	-	Bytes	-	-	-
-	Unicast Statistics	-	Packets	-	-	-
-	-	-	Bytes	-	-	-
-	Broadcast Statistics	-	Packets	-	-	-
-	-	-	Bytes	-	-	-
-	Multicast Statistics	-	Packets	-	-	-
-	-	-	Bytes	-	-	-
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	-	-	-
-	-	-	Bytes	-	-	-
Drops	-	-	Packets	5001	-	-

-	-	-		Bytes	-	-	-
-	Error	-	RX	Packets	-	-	-
-	-	-		Bytes	-	-	-
-	Error	-	TX	Packets	47	-	-
-	-	-		Bytes	-	-	-
-	No Buff	-	RX	Packets	-	-	-
-	-	-		Bytes	-	-	-
-	Traffic Statistics	-		Packets	5001	-	-
68565	-	-		Bytes	490098	-	-
5876134	-	-		Packets	-	-	-
-	Unicast Statistics	-		Bytes	-	-	-
-	-	-		Packets	-	-	-
-	Broadcast Statistics	-		Bytes	-	-	-
-	-	-		Packets	-	-	-
-	Multicast Statistics	-		Bytes	-	-	-
-	-	-		Packets	-	-	-
-	Traffic Statistics	-		Packets	5022	-	-
68565	2	-		Bytes	512244	-	-
6243130	188	37		Packets	-	-	-
-	Unicast Statistics	-		Bytes	-	-	-
-	-	-		Packets	-	-	-
-	Broadcast Statistics	-		Bytes	-	-	-
5796	-	-		Packets	-	-	-
-	-	-		Bytes	-	-	-
-	Multicast Statistics	-		Packets	5022	-	-
62769	2	-		Bytes	-	-	-
-	-	-					
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	5001	-	-
-	-	-	Bytes	-	-	-
Error	-	RX	Packets	-	-	-
-	-	-	Bytes	-	-	-
Error	-	TX	Packets	47	-	-
-	-	-	Bytes	-	-	-
No Buff	-	RX	Packets	-	-	-
-	-	-	Bytes	-	-	-
Traffic Statistics	-	-	Packets	5001	-	-
68565	-	-	Bytes	490098	-	-
5876134	-	-	Packets	-	-	-
Unicast Statistics	-	-	Bytes	-	-	-
-	-	-	Packets	-	-	-
Broadcast Statistics	-	-	Bytes	-	-	-
-	-	-	Packets	-	-	-
Multicast Statistics	-	-	Bytes	-	-	-
-	-	-	Packets	-	-	-
-	-	-	Bytes	-	-	-
Packet Statistics:						
Ingress Forwarded Packets: 1816						
Ingress Forwarded Bytes: 185232						
Ingress Drop Packets: 1						
Ingress Drop Bytes: 102						
Egress Forwarded Packets: 0						
Egress Forwarded Bytes: 0						
Egress Drop Packets: 0						
Egress Drop Bytes: 0						

9. Statistics Show Commands

9.1. Viewing ACL Statistics

9.1.1. show bcm acl statistics

This command displays all ACL statistics.

```
supervisor@rtbrick: op> show bcm acl statistics
```

```
Rule Name: mfc1
  ACL Type: multifield_ipv4
  Ordinal: 0
  ACL_Stat_id: 4294967295
  Counter_id: 16376
  Counter_engine: 254
  Acl_handle: 0
  Packet_statistics: 1524933
  Bytes_statistics: 1530770278
  Packet_statistics_core1: 0
  Bytes_statistics_core1: 0
  bcmFieldStatBytes: 0
  bcmFieldStatPackets: 0
  bcmFieldStatGreenBytes: 1530770278
  bcmFieldStatGreenPackets: 1524933
  bcmFieldStatNotGreenBytes: 0
  bcmFieldStatNotGreenPackets: 0
  bcmFieldStatAcceptedBytes: 1530770278
  bcmFieldStatAcceptedPackets: 1524933
  bcmFieldStatAcceptedGreenBytes: 1530770278
  bcmFieldStatAcceptedGreenPackets: 1524933
  bcmFieldStatAcceptedNotGreenBytes: 0
  bcmFieldStatAcceptedNotGreenPackets: 0
  bcmFieldStatDroppedBytes: 0
  bcmFieldStatDroppedPackets: 0
  bcmFieldStatDroppedGreenBytes: 0
  bcmFieldStatDroppedGreenPackets: 0
  bcmFieldStatDroppedYellowBytes: 0
  bcmFieldStatDroppedYellowPackets: 0
  bcmFieldStatDroppedRedBytes: 0
  bcmFieldStatDroppedRedPackets: 0
  bcmFieldStatDroppedNotGreenBytes: 0
  bcmFieldStatDroppedNotGreenPackets: 0
  bcmFieldStatDroppedNotYellowBytes: 0
  bcmFieldStatDroppedNotYellowPackets: 0
  bcmFieldStatDroppedNotRedBytes: 0
  bcmFieldStatDroppedNotRedPackets: 0
  bcmFieldStatDiscardedYellowOctets: 0
  bcmFieldStatDiscardedYellowPackets: 0
  bcmFieldStatDiscardedRedOctets: 0
  bcmFieldStatDiscardedRedPackets: 0
```

9.1.2. show bcm acl statistics type <type>

This command displays ACL statistics for the specified type.

```
supervisor@rtbrick: op> show bcm acl statistics type l3v4
Rule Name: radius-srv1-v4-coa-trap
  ACL Type: l3v4
  Ordinal: 0
  ACL_Stat_id: 4294967295
  Counter_id: 0
  Counter_engine: 254
  Acl_handle: 0
  Packet_statistics: 58664
  Bytes_statistics: 4145806
  Packet_statistics_core1: 0
  Bytes_statistics_core1: 0
  bcmFieldStatBytes: 0
  bcmFieldStatPackets: 0
  bcmFieldStatGreenBytes: 4145806
  bcmFieldStatGreenPackets: 58664
  bcmFieldStatNotGreenBytes: 0
  bcmFieldStatNotGreenPackets: 0
  bcmFieldStatAcceptedBytes: 4145806
  bcmFieldStatAcceptedPackets: 58664
  bcmFieldStatAcceptedGreenBytes: 4145806
  bcmFieldStatAcceptedGreenPackets: 58664
  bcmFieldStatAcceptedNotGreenBytes: 0
  bcmFieldStatAcceptedNotGreenPackets: 0
  bcmFieldStatDroppedBytes: 0
  bcmFieldStatDroppedPackets: 0
  bcmFieldStatDroppedGreenBytes: 0
  bcmFieldStatDroppedGreenPackets: 0
  bcmFieldStatDroppedYellowBytes: 0
  bcmFieldStatDroppedYellowPackets: 0
  bcmFieldStatDroppedRedBytes: 0
  bcmFieldStatDroppedRedPackets: 0
  bcmFieldStatDroppedNotGreenBytes: 0
  bcmFieldStatDroppedNotGreenPackets: 0
  bcmFieldStatDroppedNotYellowBytes: 0
  bcmFieldStatDroppedNotYellowPackets: 0
  bcmFieldStatDroppedNotRedBytes: 0
  bcmFieldStatDroppedNotRedPackets: 0
  bcmFieldStatDiscardedYellowOctets: 0
  bcmFieldStatDiscardedYellowPackets: 0
  bcmFieldStatDiscardedRedOctets: 0
  bcmFieldStatDiscardedRedPackets: 0
```

9.1.3. show bcm acl statistics type l3v4 rule <rule-name>

This command displays ACL statistics for the specified type and rule-name.

```
supervisor@rtbrick: op> show bcm acl statistics type l3v4 rule radius-srv1-  
v4-auth-trap
```

```
Rule Name: radius-srv1-v4-auth-trap  
  ACL Type: l3v4  
  Ordinal: 0  
  ACL_Stat_id: 4294967295  
  Counter_id: 0  
  Counter_engine: 254  
  Acl_handle: 0  
  Packet_statistics: 58768  
  Bytes_statistics: 4153086  
  Packet_statistics_core1: 0  
  Bytes_statistics_core1: 0  
  bcmFieldStatBytes: 0  
  bcmFieldStatPackets: 0  
  bcmFieldStatGreenBytes: 4153086  
  bcmFieldStatGreenPackets: 58768  
  bcmFieldStatNotGreenBytes: 0  
  bcmFieldStatNotGreenPackets: 0  
  bcmFieldStatAcceptedBytes: 4153086  
  bcmFieldStatAcceptedPackets: 58768  
  bcmFieldStatAcceptedGreenBytes: 4153086  
  bcmFieldStatAcceptedGreenPackets: 58768  
  bcmFieldStatAcceptedNotGreenBytes: 0  
  bcmFieldStatAcceptedNotGreenPackets: 0  
  bcmFieldStatDroppedBytes: 0  
  bcmFieldStatDroppedPackets: 0  
  bcmFieldStatDroppedGreenBytes: 0  
  bcmFieldStatDroppedGreenPackets: 0  
  bcmFieldStatDroppedYellowBytes: 0  
  bcmFieldStatDroppedYellowPackets: 0  
  bcmFieldStatDroppedRedBytes: 0  
  bcmFieldStatDroppedRedPackets: 0  
  bcmFieldStatDroppedNotGreenBytes: 0  
  bcmFieldStatDroppedNotGreenPackets: 0  
  bcmFieldStatDroppedNotYellowBytes: 0  
  bcmFieldStatDroppedNotYellowPackets: 0  
  bcmFieldStatDroppedNotRedBytes: 0  
  bcmFieldStatDroppedNotRedPackets: 0  
  bcmFieldStatDiscardedYellowOctets: 0  
  bcmFieldStatDiscardedYellowPackets: 0  
  bcmFieldStatDiscardedRedOctets: 0  
  bcmFieldStatDiscardedRedPackets: 0
```

9.2. Viewing PPPOE statistics

9.2.1. show bcm pppoe counters

This command displays all PPOE counters.


```
supervisor@rtbrick: op> show bcm pppoe counters
Subscriber-Id: 72339069014638594
  PPP SessionId: 1
  IFP Name: n/a
  Local MAC: b8:6a:97:a5:98:03
  Subscriber MAC: 00:15:01:00:00:01
  Rx_Accepted_pkts: 11
  Rx_Accepted_bytes: 789
  Rx_Dropped_pkts: 0
  Rx_Dropped_bytes: 0
  Tx_Accepted_pkts: 11
  Tx_Accepted_bytes: 529
  Tx_Dropped_pkts: 0
  Tx_Dropped_bytes: 0
Subscriber-Id: 72339069014638596
  PPP SessionId: 3
  IFP Name: n/a
  Local MAC: b8:6a:97:a5:98:03
  Subscriber MAC: 00:13:01:00:00:01
  Rx_Accepted_pkts: 206868553
  Rx_Accepted_bytes: 206855911805
  Rx_Dropped_pkts: 2
  Rx_Dropped_bytes: 2000
  Tx_Accepted_pkts: 0
  Tx_Accepted_bytes: 0
  Tx_Dropped_pkts: 0
  Tx_Dropped_bytes: 0
```

9.2.2. show bcm pppoe counters <subscriber-ID>

This command displays PPPOE counters for the specified subscriber-ID.

```
supervisor@rtbrick: op> show bcm pppoe counters 72339069014638594
Subscriber-Id: 72339069014638594
  PPP SessionId: 1
  IFP Name: n/a
  Local MAC: b8:6a:97:a5:98:03
  Subscriber MAC: 00:15:01:00:00:01
  Rx_Accepted_pkts: 11
  Rx_Accepted_bytes: 789
  Rx_Dropped_pkts: 0
  Rx_Dropped_bytes: 0
  Tx_Accepted_pkts: 11
  Tx_Accepted_bytes: 529
  Tx_Dropped_pkts: 0
  Tx_Dropped_bytes: 0
```

10. Clear Commands

10.1. Interface Counters

10.1.1. clear interface statistics

This command clears VPP interface statistics (physical/Logical)

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

10.1.2. clear interface statistics <interface-name>

This command clears VPP interface statistics (physical/Logical) for the specified interface name.

```
supervisor@rtbrick: op> clear interface statistics ifp-0/0/54
```

10.2. Clearing ACL Counters

10.2.1. clear acl counters

This command clears all ACL counters.

```
supervisor@rtbrick: op> clear acl counters
```

10.2.2. clear acl counters type <type>

This command clears the ACL counters for the specified type.

```
supervisor@rtbrick: op> clear acl counters type 13v6
```

10.2.3. clear acl counters type <type> rule <rule-name>

```
supervisor@rtbrick: op> clear acl counters type 12 rule rule1
```

10.3. BCM clear Counter

10.3.1. Interface Counters

10.3.1.1. clear bcm interface statistics

This command clears all bcm interface statistics

```
supervisor@rtbrick: op> clear bcm interface statistics
```

10.3.1.2. clear bcm interface statistics <interface-name>

This command clears interface statistics for the specified interface.

```
supervisor@rtbrick: op> clear bcm interface statistics ifp-0/0/10
```