OpenSwitch User Guide

OpenSwitch User Guide

Table of Contents

Chapter 1. Management And Utility

1.1. Management Interface

The primary goal of the management module is to facilitate the management of the device. It provides the following:

- Device access and configuration
- Event collection for monitoring, analysis, and correlation
- Device and user authentication, authorization, and accounting
- Device time synchronization
- Device image downloading

The device is configured or monitored through the management interface. All management traffic such as device ssh, tftp and so on, goes through the management interface.

1.1.1. Prerequisites

- A physical interface of the switch designated as the management interface must be specified in the *image.manifest* hardware description file.

1.1.2. Setting up the basic configuration

Configure the mode in which the management interface is going to operate. Select one of the following options:

- DHCP mode - The DHCP Client automatically populates all the management interface parameters.
- Static mode - The user manually configures the management interface parameters.

Setting up the optional configuration

1. Configure the IPv4 or the IPv6 configuration depending on the requirement.
2. Configure the secondary nameserver if fallback is required.

Verifying the configuration

1. Verify the configured values using the show interface mgmt command.
2. Verify the configuration using the show running-config command.

1.1.3. Troubleshooting the configuration

Troubleshoot the device only through the management interface. if there is any problem with the management interface configuration, you may not be able to access the device over the network. Try accessing it over the serial console.

Scenario 1

Condition

The values configured are not displayed with the show command.

Cause

The configured values will not appear in the show command if the configuration fails at the management interface daemon.

Remedy

- Check the syslog for error message.
- Check for errors in the daemon using the command systemctl status mgmt-intf -l.
- Restart the management interface daemon using the command systemctl restart mgmt-intf.

Scenario 2

Condition

Values configured in the CLI are not configured in the interface.

Cause

The management interface daemon could have crashed.

Remedy

- Check the syslog for error message.
- Check for errors in the daemon using the command systemctl status mgmt-intf -l.
- Restart the management interface daemon using the command systemctl restart mgmt-intf.

Scenario 3

Condition

The mode is dhcp, but no DHCP attributes are populated.

Cause

The dhclient might be down.

Remedy

- Check the syslog for error message.
- Check for errors in the daemon using the command systemctl status mgmt-intf -l.
- Restart the management interface daemon using the command systemctl restart mgmt-intf.

Scenario 4

Condition

The mode is dhcp, but IPv6 attributes are not seen in the show interface mgmt command.

Cause

Currently IPv6 notifications are not available.

Remedy

- The user can start the shell using "start-shell" command and check the IPv6 attributes by running the command ip addr show.

1.2. AAA feature

1.2.1. Overview

The AAA feature is used for authenticating users who access the switch management interface using console, SSH, or REST. The AAA feature supports the following:

- Local or RADIUS authentication.
- Configuring the RADIUS authentication type.
- Configuring RADIUS servers (maximum of 64 RADIUS servers).
- Configuring the SSH authentication method.

This feature currently supports user authentication based on user name and password.

1.2.2. How to use the feature, Scenario 1

Setting up scenario 1 basic configuration

1. Create a user on the switch, and configure a password.
2. Configure the authentication mode. Select one of the following options:
   - local
   - RADIUS
3. Configure RADIUS servers.

Setting up scenario 1 optional configuration

1. Change the default value of the shared secret used for communication between the switch and the RADIUS server.
2. Change the default value of the port used for communication with the RADIUS server.
3. Change the default value of the connection retries.
4. Change the connection timeout default value.
5. Change the RADIUS authentication type to CHAP or PAP (default is PAP).

Verifying scenario 1 configuration

1. Verify the configuration using the show command.
2. Verify the configuration using the show running-config command for non default values.

1.2.3. Troubleshooting scenario 1 configuration

Case 1

Condition

The local authentication is configured, but the user is unable to log in with local password.

Possible causes

- The AAA daemon is not active. Check the AAA status with the following command: ps -ef **|** grep ops_aaautilspamcfg.
- The PAM configuration files are not present. Check for the presence of the common-*****-access files located at /etc/pam.d/.
- The wrong password has been entered.

Remedies

- If the daemon is not running, restart it by entering systemctl start aaautils.service.
- If the PAM configuration files are removed, copy the PAM configuration files from another switch located at etc/pam.d/.
- Verify that the correct password has been entered.
- For more information, verify the auth.log file located at /var/log/auth.log.

Case 2

Condition

The RADIUS authentication is configured, but the user is unable to log in with the RADIUS server password.

Possible causes

- The AAA daemon is not active. Check the AAA status with the following command: ps -ef **|** grep ops_aaautilspamcfgg.
- The RADIUS server is not active. Verify if the RADIUS server is stopped.
- There appears to be a difference in the configuration of the RADIUS servers on the switch and on the host. Enter the show radius-server command on the switch to verify the RADIUS servers' configuration.
- The wrong password has been entered.

Remedies

- If the daemon is not running, restart it by entering systemctl start aaautils.service.
- If the PAM configuration files are removed, copy the PAM configuration files from another switch located at etc/pam.d/.
- Restart the RADIUS server.
- Verify that the correct password has been entered.

- For more information, verify the auth.log file located at /var/log/auth.log.

1.2.4. Scenario 2

Setting up scenario 2 basic configuration

1. Enable RADIUS authentication.
2. Enable fallback to local authentication.
3. Configure the RADIUS server.

Setting up scenario 2 optional configuration

1. Change the default value of the shared secret used for communication between the switch and the RADIUS server.
2. Change the default value of the port used for communication with the RADIUS server.
3. Change the default value of the connection retries.
4. Change the connection timeout default value.
5. Change the RADIUS authentication type to CHAP or PAP (default is PAP).

Verifying scenario 2 configuration

1. Verify the configuration using the show command.
2. Verify the configuration using the show running-config command for non-default values.

1.2.5. Troubleshooting scenario 2 configuration

Condition

The RADIUS server is unreachable, and the user cannot log in with local credentials even though fallback to local is enabled.

Possible causes

- The AAA daemon is not active. Check the AAA status with the following command: ps -ef | grep ops_aaautilspamcfg.
- The PAM configuration files are not present. Check for the presence of the common-**-access files located at /etc/pam.d/.
- The wrong password has been entered.

Remedies

- If the daemon is not running, restart it by entering systemctl start aaautils.service.
- If the PAM configuration files are removed, copy the PAM configuration files from another switch located at etc/pam.d/.

- Verify that the correct password has been entered.
- For more information, verify the auth.log file located at /var/log/auth.log.

1.2.6. Scenario 3

Setting up scenario 3 basic configuration

1. Create a user on the switch.
2. Enable a SSH password or a public key authentication method.

Setting up scenario 3 optional configuration

N/A

Verifying scenario 3 configuration

1. Verify the configuration using the show command.
2. Verify the configuration using the show running-config command for non-default values.

1.2.7. Troubleshooting scenario 3 configuration

Case 1

Condition

The SSH password authentication is enabled, but the user is not able to log in with the password.

Possible causes

- The AAA daemon is not active. Check the AAA status with the following command: ps -ef | grep ops_aaautilspamcfg.
- The SSH configuration file is not present. Check for the presence of the sshd_config file located at /etc/ssh/.
- The wrong password has been entered.

Remedies

- If the daemon is not running, restart it by entering systemctl start aaautils.service.
- If the SSH configuration files is removed, copy the SSH configuration file sshd_config from another switch.
- Verify that the correct password has been entered.
- For more information, verify the auth.log file located at /var/log/auth.log.

Case 2

Condition

SSH public key authentication is enabled, but the user is not able to login.

Possible causes

- The AAA daemon is not active. Check the AAA status with the following command: ps -ef | grep ops_aaautilspamcfg.
- The SSH configuration file is not present. Check for the presence of the sshd_config file located at /etc/ssh/.
- The user’s public key is not present on the switch. Check for the presence of the public key by entering something similar to the following with the user’s information: /home/<user>/.ssh/id_rsa.pub

Remedies

- If the daemon is not running, restart it by entering systemctl start aaautils.service.
- If the SSH configuration files is removed, copy the SSH configuration file sshd_config from another switch.
- Copy the public key manually to the switch by entering something similar with the user’s information as follows: /home/<user>/.ssh/id_rsa.pub
- For more information, verify the auth.log file located at /var/log/auth.log.

1.3. TACACS

TACACS+ is a protocol that handles authentication, authorization, and accounting (AAA) services. TACACS+ client functionality is supported on the switch.

1.3.1. Prerequisites

- A TACACS+ server (either local or remote) is needed for AAA services.
- OpenSwitch needs to have management interface UP and enabled.

1.3.2. Limitations

- A maximum of 64 TACACS+ servers can be configured.
- Server can be configured with a unicast IPV4/IPV6 address or FQDN.
- A maximum of 28 user-defined AAA servers-groups can be configured.
- Session-type (console/ssh/telnet) configuration provided together as *default* configuration for authentication.
- TACACS+ server reachability is over the management interface.

1.3.3. Defaults

- The default authentication tcp-port is 49.
- The default authentication timeout value is five.
- The default authentication key (shared-secret between client and server) is testing123-1.
- The default authentication-protocol is pap.

1.4. DHCP Relay

1.4.1. Overview

The Dynamic Host Configuration Protocol (DHCP) is used for configuring hosts with IP addresses and other configuration parameters, without human intervention. The protocol is composed of three components: the DHCP client, the DHCP server, and the DHCP relay agent. The DHCP client sends broadcast request packets to the network. DHCP servers respond with broadcast packets that offer IP parameters, such as an IP address for the client. After the client chooses the IP parameters, communication between the client and the server is by unicast packets. The function of the DHCP relay agent is to forward the DHCP messages to other subnets so that the DHCP server does not have to be on the same subnet as the DHCP clients. The DHCP relay agent transfers DHCP messages from the DHCP clients located on a subnet without a DHCP server, to other subnets. It also relays answers from DHCP servers to DHCP clients. The DHCP relay agent on the routing switch forwards DHCP client packets to all DHCP servers (helper IP addresses) that are configured in the table administered for each interface. The helper address configuration is allowed only on data plane interfaces. The helper address should not be multicast or loopback address.

DHCP relay option 82

Option 82 is called the relay agent information option. The option 82 field is inserted/replaced or the packet with this option is dropped by the DHCP relay agent, when forwarding client-originated DHCP packets to a DHCP server. Servers recognizing the relay agent information option may use the information to implement an IP address or other parameter assignment policies. The relay agent relays the server-to-client replies to the client.

Hop count in DHCP requests

When a DHCP client broadcasts requests, the DHCP relay agent in the routing switch receives the packets and forwards them to the DHCP server (on a different subnet, if necessary.) During this process, the DHCP relay agent increments the hop count before forwarding DHCP packets to the server. The DHCP server, in turn, includes the hop count in DHCP header from the received DHCP request in the response sent back to a DHCP client. This is enabled by default.

Configuring a BOOTP/DHCP relay gateway

The DHCP relay agent selects the lowest-numbered IP address on the interface to use for DHCP messages. The DHCP server then uses this IP address when it assigns client addresses. However, this IP address may not be the same subnet as the one on which the client needs the DHCP service. This feature provides a way to configure a gateway address for the DHCP relay agent to use for relayed DHCP requests, rather than the DHCP relay agent automatically assigning the lowest-numbered IP address.

1.4.2. Configure DHCP relay

Helper address configuration on an interface is allowed even if routing is disabled on the interface, but DHCP relay functionality will be inactive on that interface. In case a client has received an IP address, and no routing is configured, the IP address is valid on the client until the lease time expires.

Syntax [no] dhcp-relay - Enable/Disable dhcp-relay. By default, it is enabled.

Syntax [no] ip helper-address <IPv4-address> - Configure the IP helper-address needed by DHCP relay on a particular interface.

Explanation of parameters

- IPv4-address - The IPv4 address of the protocol server. This is a unicast address of a destination server on another subnet. The maximum number of helper addresses that can be configured per interface is eight. DHCP relay functions on L3 interfaces that include split-interfaces and sub-interfaces.

Syntax show ip helper-address [interface <interface-name>] - Displays the configured IP helper-address(es).

Explanation of parameters

- interface <interface-name> - The interface on which server addresses are configured.

1.4.3. How to use DHCP relay

Example 1

```
switch# configure terminal
switch(config)# dhcp-relay
switch# show dhcp-relay
```

```
DHCP Relay Agent                 : Enabled
DHCP Request Hop Count Increment : Enabled
Option 82                        : Disabled
Response Validation              : Disabled
Option 82 Handle Policy          : replace
Remote ID                        : mac
```

```
DHCP Relay Statistics:
```

```
Client Requests       Server Responses
```

```
Valid      Dropped    Valid      Dropped
---------- ---------- ---------- ----------
60         10         60         10
```

```
DHCP Relay Option 82 Statistics:
```

```
Client Requests       Server Responses
```

```
Valid      Dropped    Valid      Dropped
---------- ---------- ---------- ----------
50         8          50         8
```

Example 2

```
switch# configure terminal
switch(config)# interface 1
switch(config-if)# ip helper-address 192.168.10.1
switch(config-if)# ip helper-address 192.168.20.1
switch(config-if)# ip helper-address 192.168.30.1
```

```
switch# show ip helper-address
IP Helper Addresses
```

```
Interface: 1
 IP Helper Address
 -----------------
 192.168.10.1
 192.168.20.1
 192.168.30.1
```

Example 3

```
switch# configure terminal
switch(config)# interface 1
switch(config-if)# no ip helper-address 192.168.10.1
switch(config-if)# no ip helper-address 192.168.20.1
switch(config-if)# no ip helper-address 192.168.30.1
switch# show ip helper-address
No helper-address configuration found.
```

Example 4

```
switch# configure terminal
switch(config)# no dhcp-relay
switch# show dhcp-relay
DHCP Relay Agent                 : Disabled
DHCP Request Hop Count Increment : Enabled
Option 82                        : Disabled
Response Validation              : Disabled
Option 82 Handle Policy          : replace
Remote ID                        : mac
```

```
DHCP Relay Statistics:
```

```
Client Requests       Server Responses
Valid      Dropped    Valid      Dropped
---------- ---------- ---------- ----------
60         10         60         10
DHCP Relay Option 82 Statistics:
Client Requests       Server Responses
Valid      Dropped    Valid      Dropped
---------- ---------- ---------- ----------
50         8          50         8
```

1.4.4. Configure DHCP relay option 82

Configure dhcp-relay option 82 globally:

Syntax dhcp-relay option 82 < replace [validate] | drop [validate] | keep | validate [replace | drop] > [ip | mac]

Disable option 82 completely:

Syntax no dhcp-relay option 82

Disable response validation only for drop/replace policy. Not applicable if keep policy was selected.

Syntax no dhcp-relay option 82 [validate]

Display the dhcp-relay option 82 configurations:

Syntax show dhcp-relay

Explanation of parameters

- drop - Configures the router to unconditionally drop any client DHCP packet received with existing option 82 fields. This means that such packets are not forwarded. Use this option where access to the router by untrusted clients is possible.
- keep - For any client DHCP packet received with existing option 82 fields, configures the router to forward the packet as-is, without replacing or adding to the existing option 82 fields.
- replace - Configures the switch to replace existing option 82 fields in an inbound client DHCP packet with an Option 82 field for the switch.
- validate - Operates when the routing switch is configured with append, replace, or drop as a forwarding policy. With validate enabled, the routing switch applies stricter rules to an incoming Option 82 server response to determine whether to forward or drop the response.
- ip - Specifies the IP address of the interface on which the client DHCP packet enters the switch.
- mac - Specifies the MAC address of the router. (The MAC address used is the same MAC address that is assigned to all interfaces configured on the router.) This is the default setting.

1.4.5. How to use DHCP relay option 82

Example 1

```
switch# configure terminal
switch(config)# dhcp-relay option 82 replace validate mac
switch# show dhcp-relay
DHCP Relay Agent : Enabled
DHCP Request Hop Count Increment : Enabled O
ption 82 : Enabled
Response Validation : Enabled
Option 82 Handle Policy : replace
Remote ID : mac
```

```
DHCP Relay Statistics:
Client Requests Server Responses
Valid Dropped Valid Dropped
60 10 60 10
DHCP Relay Option 82 Statistics:
```

```
Client Requests Server Responses
Valid Dropped Valid Dropped
50 8 50 8
```

Example 2

```
switch# configure terminal
switch(config)# no dhcp-relay option 82
switch# show dhcp-relay
DHCP Relay Agent : Enabled
DHCP Request Hop Count Increment : Enabled
Option 82 : Disabled
Response Validation : Enabled
Option 82 Handle policy : replace
Remote ID : mac
```

```
DHCP Relay Statistics:
Client Requests Server Responses
Valid Dropped Valid Dropped
60 10 60 10
DHCP Relay Option 82 Statistics:
Client Requests Server Responses
Valid Dropped Valid Dropped
50 8 50 8
```

1.4.6. Configure DHCP relay BOOTP gateway

Syntax [no] ip bootp-gateway <IPv4-address>

Configure the IP bootp-gateway needed by DHCP relay on a particular interface.

Explanation of parameters

- IPv4-address - The IPv4 address of the gateway. Provides a way to configure a gateway address for the DHCP relay agent to use for DHCP requests, rather than the DHCP relay agent automatically assigning the lowest-numbered IP address.

Syntax show dhcp-relay bootp-gateway [interface <interface-name>]

Displays the bootp-gateway configuration.

Explanation of parameters

- interface <`interface-name`> - Interface on which gateway address is configured.

1.4.7. How to use DHCP relay BOOTP gateway

Example 1

```
switch# configure terminal
switch(config)# interface 1
switch(config-if)# ip bootp-gateway 1.1.1.1
```

```
switch# show dhcp-relay bootp-gateway
BOOTP Gateway Entries   Interface BOOTP Gateway
1 1.1.1.1
```

Example 2

```
switch# configure terminal
switch(config)# interface 1
switch(config-if)# no ip bootp-gateway 1.1.1.1
switch# show dhcp-relay bootp-gateway
No bootp-gateway configuration found.
```

1.4.8. Configure DHCP relay hop count increment

Syntax [no] dhcp-relay hop-count-increment

- Enable/Disable the dhcp-relay hop-count-increment command. By default, it is enabled.*

Syntax show dhcp-relay

- Displays the dhcp-relay hop-count-increment configurations.*

1.4.9. How to use DHCP relay hop count increment

Example 1

```
switch# configure terminal
switch(config)# dhcp-relay hop-count-increment
switch# show dhcp-relay
DHCP Relay Agent : Enabled
DHCP Request Hop Count Increment : Enabled
Option 82 : Disabled
Response Validation : Disabled
Option 82 Handle Policy : replace
Remote ID : mac
```

```
DHCP Relay Statistics:
Client Requests Server Responses
Valid Dropped Valid Dropped
60 10 60 10
DHCP Relay Option 82 Statistics:
Client Requests Server Responses
Valid Dropped Valid Dropped
50 8 50 8
```

Example 2

```
switch# configure terminal
switch(config)# no dhcp-relay hop-count-increment
switch# show dhcp-relay
DHCP Relay Agent : Enabled
DHCP Request Hop Count Increment : Disabled
```

```
Option 82 : Disabled Response
Validation : Disabled
Option 82 Handle Policy : replace
Remote ID : mac
```

```
DHCP Relay Statistics:
Client Requests Server Responses
Valid Dropped Valid Dropped
60 10 60 10
DHCP Relay Option 82 Statistics:
Client Requests Server Responses
Valid Dropped Valid Dropped
50 8 50 8
```

1.5. DHCP-TFTP server

1.5.1. Overview

This guide provides details for configuring the DHCP-TFTP server that is present in the switch. All DHCP configurations work in the dhcp-server context. All TFTP configurations work in the tftp-server context.

1.5.2. Prerequisites

All of the interfaces on which the DHCP-TFTP server listens must be administratively up. To enable a DHCP server, at least one dynamic range configuration must be set. Both the interface IP address and the IP addresses range configured for DHCP clients must be in the same subnet. To configure a static IP address for requests coming to an interface, a dynamic range configuration must be previously set for that interface, and the static IP address and dynamic IP addresses range must be in the same subnet.

1.5.3. Configuring a DHCP server

Changing to dhcp-server context

The dhcp-server command changes the configure terminal context to the dhcp-server context.

```
switch# configure terminal
switch(config)# dhcp-server
switch(config-dhcp-server)#
```

Setting the dynamic range configuration

The **range <range-name> start-ip-address(<ipv4_address> | <ipv6_address>) end-ip-address (<ipv4_address> | <ipv6_address>) netmask <subnet_mask> broadcast <broadcast_address> match tags <match_tag_names> set tag <set_tag_name> prefix-len <prefix_length_value> lease-duration <lease_duration_value>** static command sets the dynamic range configuration for the DHCP server. Each dynamic range configuration should have a unique name.

- Parameter start-ip-address sets the first ip address of the dynamic range.
- Parameter end-ip-address sets the last ip address of the dynamic range.
- Parameters start-ip-addressand end-ip-address are not optional.
- Parameter netmask sets the subnet mask.
- Parameter broadcast sets the broadcast address for the range specified, but it must not be set without setting netmask.
- Parameter prefix-len sets the prefix length for IPv6 address range. Either IPv4 or IPv6 address can be set for the dynamic range.
- Paramaters netmask and broadcast should not be specified for IPv6 address range while parameter prefix-len must not be set for the IPv4 address range.

- Parameter set tags sets alphanumeric labels which marks networks so that dhcp options may be specified on a per-network basis. Only one tag should be specified for this parameter.
- Parameter match-tags sets the matching labels. Mutliple tags can be set for this parameter.
- Parameter lease-duration sets the lease time. If this parameter is not specified, default value of 60 minutes is set.

```
switch(config-dhcp-server)# range dynamic_1 start-ip-address 10.0.0.1
end-ip-address 10.255.255.254 netmask 255.0.0.0 broadcast 10.255.255.255
match tags tag1,tag2,tag3 set tag tag4
switch(config-dhcp-server)# range dynamic start-ip-address 10.0.0.1
end-ip-address 10.255.255.254 netmask 255.0.0.0 broadcast 10.255.255.255
match tags tag1,tag2,tag3 set tag tag4
switch(config-dhcp-server)#
switch(config-dhcp-server)# range dynamic_2 start-ip-address
2001:0db8:85a3:0000:0000:8a2e:0370:7334 end-ip-address
2001:0db8:85a3:0000:0000:8a2e:0370:7340  prefix-len 64  match tags tag6,
tag7,tag8 set tag tag5
```

To remove the dynamic configuration, use the **no range range <range-name> start-ip-address (<ipv4_address> | <ipv6_address>) end-ip-address (<ipv4_address> | <ipv6_address>) netmask <subnet_mask> broadcast <broadcast_address> match tags <match_tag_names> set tag <set_tag_name> prefix-len <prefix_length_value> lease-duration <lease_duration_value>** static command.

```
switch(config-dhcp-server)# no range dynamic_1 start-ip-address 10.0.0.1
end-ip-address 10.255.255.254 netmask 255.0.0.0 broadcast 10.255.255.255
match tags tag1,tag2,tag3 set tag tag4
switch(config-dhcp-server)# range dynamic start-ip-address 10.0.0.1
end-ip-address 10.255.255.254 netmask 255.0.0.0 broadcast 10.255.255.255
match tags tag1,tag2,tag3 set tag tag4
switch(config-dhcp-server)#
switch(config-dhcp-server)# no range dynamic_2 start-ip-address
2001:0db8:85a3:0000:0000:8a2e:0370:7334 end-ip-address
2001:0db8:85a3:0000:0000:8a2e:0370:7340  prefix-len 64  match tags tag6,
tag7,tag8 set tag tag5
```

Setting static configurations

The **static (<ipv4_address> | <ipv6_address>) match-mac-addresses <mac_addresses> match-client-hostname <hostname> match-client-id <client-id> set tags <set_tag_names> lease-duration <lease_duration_value>** command sets the static IP allocation configuration.

- Parameter static sets the ip address for the static ip address allocation.
- Parameter match-mac-addresses sets the MAC address of the client. The user can specify multiple MAC addresses for the static IP but only one MAC address should be active at any point.
- Paremeters match-client-hostname and match-client-id sets the client hostname and clied ID respectively.
- Parameters match-mac-addresses, match-client-hostname and match-client-id are optional but at lease one of these three parameters should be specified.

- Paramater set tags sets alphanumeric labels which marks networks so that dhcp options may be specified on a per-network basis.
- Parameter set tags is optional. If specified, mutliple tags can be set.
- Parameter lease-duration sets the lease time. If this parameter is not specified, default value of 60 minutes is set.

```
switch(config-dhcp-server)# static 10.0.0.5 match-mac-addresses
36:d4:1b:12:ea:52 match-client-hostname 95_h2 set tags tag1,tag2,tag3
lease-duration 65
```

To remove the static configuration, use the **no static (<ipv4_address> | <ipv6_address>) match-mac-addresses <mac_addresses> match-client-hostname <hostname> match-client-id <client-id> set tags <set_tag_names> lease-duration <lease_duration_value>** command.

```
switch(config-dhcp-server)#
switch(config-dhcp-server)# no static 10.0.0.5 match-mac-addresses
36:d4:1b:12:ea:52 match-client-hostname 95_h2 set tags tag1,tag2,tag3
lease-duration 65
```

1.5.4. Setting the DHCP options configuration

The **set option-name <option_name> option-value <option_value> match tags <match_tag_names> ipv6** command sets the DHCP options by specifying the option-name, and the **option set option-number <option_number> option-value <option_value> match tags <match_tag_names> ipv6** command sets the DHCP options by specifying the option-number.

- Parameter option-name sets the DHCP option name,
- Parameter option-number sets the DHCP option number.
- Parameter option-value sets the DHCP option value.
- Parameter match-tags sets the matching labels (optional).

The list of supported `option-name` and corresponding `option-number` for IPv4 are:

```
| option-name              | option-number |
|--------------------------|---------------|
| netmask                  | 1             |
| time-offset              | 2             |
| router                   | 3             |
| dns-server               | 6             |
| log-server               | 7             |
| lpr-server               | 9             |
| hostname                 | 12            |
| boot-file-size           | 13            |
| domain-name              | 15            |
| swap-server              | 16            |
| root-path                | 17            |
| extension-path           | 18            |
| ip-forward-enable        | 19            |
| non-local-source-routing | 20            |
| policy-filter            | 21            |
| max-datagram-reassembly  | 22            |
| default-ttl              | 23            |
| mtu                      | 26            |
| all-subnets-local        | 27            |
| broadcast                | 28            |
| router-discovery         | 31            |
| router-solicitation      | 32            |
| static-route             | 33            |
| trailer-encapsulation    | 34            |
| arp-timeout              | 35            |
| ethernet-encap           | 36            |
| tcp-ttl                  | 37            |
| tcp-keepalive            | 38            |
| nis-domain               | 40            |
| nis-server               | 41            |
| ntp-server               | 42            |
| vendor-encap             | 43            |
| netbios-ns               | 44            |
| netbios-dd               | 45            |
| netbios-nodetype         | 46            |
| netbios-scope            | 47            |
| x-windows-fs             | 48            |
| x-windows-dm             | 49            |
| requested-address        | 50            |
| lease-time               | 51            |
| option-overload          | 52            |
| message-type             | 53            |
| server-identifier        | 54            |
| parameter-request        | 55            |
| message                  | 56            |
| max-message-size         | 57            |
| T1                       | 58            |
| T2                       | 59            |
| vendor-class             | 60            |
| client-id                | 61            |
| nis+-domain              | 64            |
| nis+-server              | 65            |
| tftp-server              | 66            |
| bootfile-name            | 67            |
| mobile-ip-home           | 68            |
| smtp-server              | 69            |
| pop3-server              | 70            |
| nntp-server              | 71            |
| irc-server               | 74            |
| user-class               | 77            |
| FQDN                     | 81            |
| agent-id                 | 82            |
| client-arch              | 93            |
| client-interface-id      | 94            |
| client-machine-id        | 97            |
| subnet-select            | 118           |
| domain-search             | 119           |
| sip-server                | 120           |
| classless-static-route    | 121           |
| vendor-id-encap           | 125           |
| server-ip-address         | 255           |
```

The list of supported `option-name` and corresponding `option-number` for IPv6 are:

```
| option-name               | option-number |
|---------------------------|---------------|
| client-id                 | 1             |
| server-id                 | 2             |
| ia-na                     | 3             |
| ia-ta                     | 4             |
| iaaddr                    | 5             |
| oro                       | 6             |
| preference                | 7             |
| unicast                   | 12            |
| status                    | 13            |
| rapid-commit              | 14            |
| user-class                | 15            |
| vendor-class              | 16            |
| vendor-opts               | 17            |
| sip-server-domain         | 21            |
| sip-server                | 22            |
| dns-server                | 23            |
| domain-search             | 24            |
| nis-server                | 27            |
| nis+-server               | 28            |
| nis-domain                | 29            |
| nis+-domain               | 30            |
| sntp-server               | 31            |
| information-refresh-time  | 32            |
| FQDN                      | 39            |
| ntp-server                | 56            |
| bootfile-url              | 59            |
| bootfile-param            | 60            |
```

Refer to RFC 2132 for the format of a valid `option-value` parameter for the corresponding `option-name/option-number` options.

```
switch# configure terminal
switch(config)# dhcp-server switch(config-dhcp-server)
#option set option-name Router option-value 10.0.0.1 match tags
tag1,tag2,tag3
switch(config-dhcp-server)#option set option-number 3 option-value 10.0.0.1
match tags tag1,tag2,tag3
```

To remove the DHCP options configuration, use the `**no option set option-name <option_name> option-value <option_value> match tags <match_tag_names> ipv6** and **no option set option-number <option_number> option-value <option_value> match tags <match_tag_names> ipv6** commands.

```
switch# configure terminal
switch(config)# dhcp-server
switch(config-dhcp-server)# no option set option-name Router option-value
10.0.0.1 match tags tag1,tag2,tag3
switch(config-dhcp-server)# no option set option-number 3 option-value
10.0.0.1 match tags tag1,tag2,tag3
```

1.5.5. Setting the DHCP match configuration

The `match set tag <set_tag_name> match-option-name <option_name> match-option-value <option_value>` and `match set tag <set_tag_name> match-option-number <option_number> match-option-value <option_value>` commands set the configuration for the server to set the tag if the client sends a DHCP option of the given number or name.

- Parameter `match-option-name` sets the option name to be matched in the client request.
- Parameter `match-option-value` sets the option number to be matched in the client request.
- Parameter `match-option-value` is optional. If this parameter is not specified, this command sets the tag if the client sends a DHCP option of the given name. If the option value is specified, then the tag would be set only if the option is sent and matches the value.

```
switch# configure terminal
switch(config)# dhcp-server
switch(config-dhcp-server)# match set tag tag1 match-option-name Router
match-option-value 10.0.0.1
switch(config-dhcp-server)# match set tag tag1 match-option-number 3
match-option-value 10.0.0.1
```

To remove the DHCP match configuration, use the no `match set tag <set_tag_name> match-option-name <option_name> match-option-value <option_value>` and no `match set tag <set_tag_name> match-option-number <option_number> match-option-value <option_value>` commands.

switch# configure terminal switch(config)# dhcp-server switch(config-dhcp-server)# no match set tag tag1 match-option-name Router match-option-value 10.0.0.1 switch(config-dhcp-server)# no match set tag tag1 match-option-number 3 match-option-value 10.0.0.1

1.5.6. Setting the DHCP BOOTP configuration

The `boot set file <file_name> match tag <match_tag_name>` command sets the BOOTP options that are returned by the DHCP server.

- Parameter `file` sets the file name.
- Optional parameter `match tag` sets the matching labels.

```
switch# configure terminal
switch(config)# dhcp-server
switch(config-dhcp-server)# boot set file /tmp/tftp_file match tag tag1
```

To remove the DHCP BOOTP configuration use the no `boot set file <file_name> match tag <match_tag_name>` command.

```
switch# configure terminal
switch(config)# dhcp-server
switch(config-dhcp-server)# no boot set file /tmp/tftp_file match tag tag1
```

1.5.7. Viewing DHCP server configuration information

The `show dhcp-server` command displays information about the DHCP server configuration. The information includes the details of dynamic, static, options, match, and BOOTP configuration.

```
switch# show dhcp-server
DHCP dynamic IP allocation configuration
Name Start IP Address End IP Address Netmask Broadcast
dynamic 10.0.0.1 10.255.255.254 255.0.0.0 10.255.255.255
DHCP static IP allocation configuration
IP Address Hostname Lease time MAC-Address Set tags
10.0.0.25 95_h2 65 36:d4:1b:12:ea:52 tag1,tag2,tag3
DHCP options configuration
Option Number Option Name Option Value ipv6 Match tags
3 * 10.0.0.1 False tag1,tag2,tag3
```

```
Router            10.0.0.1              False  tag1,tag2,tag3
```

```
DHCP Match configuration
Option Number Option Name Option Value Set tag
```

```
3 * 10.0.0.1 tag1
```

```
            Router            10.0.0.1              tag1
 DHCP BOOTP configuration
Tag File
tag1 /tmp/tftp_file
```

1.5.8. Viewing DHCP server leases information

The `show dhcp-server leases` command displays information about the DHCP server leases database. The database is updated by the DHCP server whenever an IP address is assigned to the client and a lease entry is expired or modified. The information includes the IP address, MAC address, lease expiry time, the client hostname, and the client ID.

```
switch# show dhcp-server leases
Expiry Time MAC Address IP Address Hostname and Client-id
Wed Sep 23 23:07:12 2015 df:36:12:1b:54:ea 10.0.0.5 95_h1 *
Wed Sep 23 22:05:10 2015 36:d4:1b:12:ea:52 10.0.0.25 95_h2 *
```

1.6. Configuring a TFTP server

1.6.1. Changing to tftp-server context

The `tftp-server` command changes the configure terminal context to the tftp-server context.

```
switch# configure terminal
switch(config)# tftp-server
switch(config-tftp-server)#
```

1.6.2. Setting the TFTP server configuration

The `enable` command enables the TFTP server.

```
switch# configure terminal
switch(config)# tftp-server
switch(config-tftp-server)# enable
```

To disable the TFTP server, use the `no enable` command.

```
switch# configure terminal
switch(config)# tftp-server
switch(config-tftp-server)# no enable
```

1.6.3. Setting the TFTP server secure mode configuration

The `secure-mode` command enables the TFTP server secure mode.

```
switch# configure terminal
switch(config)# tftp-server
switch(config-tftp-server)# secure-mode
```

To disable TFTP server secure mode, use the `no secure-mode` command.

```
switch# configure terminal
switch(config)# tftp-server
switch(config-tftp-server)# no secure-mode
```

1.6.4. Setting the TFTP server root path configuration

The `path <path_name>` command sets the TFTP root path configuration. Only absolute paths (starting with /) are allowed, and relative paths are not allowed.

```
switch# configure terminal
switch(config)# tftp-server
switch(config-tftp-server)# path /tmp/
```

1.6.5. Viewing the TFTP server information

The `show tftp-server` command displays information about the TFTP server configuration. The information includes details of the TFTP server, secure mode, and the path configuration.

```
switch# show tftp-server
TFTP server configuration
TFTP server : Enabled
TFTP server secure mode : Enabled
TFTP server file path : /tmp/
```

1.7. SFTP Utility

1.7.1. Overview

The SFTP (Secure File Transfer Protocol) command is a very common method for transferring a file or executing a command in a secure mode. SFTP makes use of encrypted SSH session for it’s operation. It provides an alternative to TFTP (Trivial File Transfer Protocol) for transferring sensitive information to and from the switch. Files transferred using SFTP are encryted and require authentication, thus providing greater security to the switch. Both SFTP server and client functionality is supported in OpenSwitch. If switch acts as an SFTP server, it listens on a default SSH port (default SSH port is 22) for any incoming connections. As an SFTP client, it can initiate a file transfer or enter remote device. SFTP client commands can be accessed only by an admin user. The SFTP service is supported only in management interface.

1.7.2. SFTP server

Syntax [no] sftp server enable Enables/Disables SFTP server. By default, it is disabled.

Syntax show sftp server Display the SFTP server status.

1.7.3. How to use SFTP server

Example

```
switch(config)#sftp server enable
switch#show sftp server
SFTP server configuration
........................................
SFTP server : Enabled
switch#show running-config
Current configuration:
!
!
!
sftp server enable
switch(config)#no sftp server enable
switch#show sftp server
SFTP server configuration
........................................
SFTP server : Disabled
```

1.7.4. SFTP client

Syntax copy sftp WORD (ipv4-address | hostname | ipv6-address) WORD [WORD]

- get operation

Syntax copy sftp WORD (ipv4-address | hostname | ipv6-address)

- get operation
- put operation

1.7.5. How to use SFTP client

```
switch# copy sftp user hostname srcpath
```

- Provide the username, hostname and the source file path of the remote device. Default destination is */var/local/*

```
switch# copy sftp abc hostmachine source-file
abc@hostmachine's password:
Connected to 10.1.1.1.
Fetching source-file to source-file
source-file                   100%   25     0.0KB/s   00:00
switch#
```

```
switch# copy sftp user IPv4-address srcpath
```

- Provide the username, IPv4 address and the source file path of the remote device. Default destination is */var/local/*

```
switch# copy sftp abc 10.1.1.1 source-file
abc@10.1.1.1's password:
Connected to 10.1.1.1.
Fetching source-file to source-file
source-file                   100%   25     0.0KB/s   00:00
switch#
```

```
switch# copy sftp user IPv6-address srcpath
```

- Provide the username, IPv6 address and the source file path of the remote device. Default destination is */var/local/*

```
switch# copy sftp abc a::1 source-file
abc@a::1's password:
Connected to a::1.
Fetching source-file to source-file
source-file                   100%   25     0.0KB/s   00:00
switch#
```

```
switch# copy sftp user hostname srcpath dstpath
```

- Provide the username, hostname, source path of the remote device and the destination path where the file is to be stored on the local device.

```
switch# copy sftp abc hostmachine source-file destination-file
abc@hostmachine's password:
Connected to hostmachine.
Fetching source-file to destination-file
source-file                   100%   25     0.0KB/s   00:00
switch#
```

```
switch# copy sftp user IPv4-address srcpath dstpath
```

- Provide the username, IPv4 address, source path of the remote device and the destination path where the file is to be stored on the local device.

```
switch# copy sftp abc 10.1.1.1 source-file destination-file
abc@10.1.1.1's password:
Connected to 10.1.1.1.
Fetching source-file to destination-file
source-file                   100%   25     0.0KB/s   00:00
switch#
```

```
switch# copy sftp user IPv6-address srcpath dstpath
```

- Provide the username, IPv6 address, source path of the remote device and the destination path where the file is to be stored on the local device.

```
switch# copy sftp abc a::1 source-file destination-file
abc@a::1's password:
Connected to a::1.
Fetching source-file to destination-file
source-file                   100%   25     0.0KB/s   00:00
switch#
```

```
switch# copy sftp user hostname
```

- Provide the username and hostname of the remote device.

```
switch# copy sftp abc hostmachine
abc@hostmachine's password:
Connected to hostmachine.
sftp>
sftp> get /users/abc/test_file
Fetching /users/abc/test_file to test_file
/users/abc/test_file      100%  212     0.2KB/s   00:00
sftp> put test_file /users/abc/
Uploading test_file to /users/abc/test_file
test_file     100%  212     0.2KB/s   00:00
```

```
switch# copy sftp user IPv4-address
```

- Provide the username and IPv4 address of the remote device

```
switch# copy sftp abc 10.1.1.1
The authenticity of host '10.1.1.1 (10.1.1.1)' can't be established.
ECDSA key fingerprint is SHA256:uWeyXm2j6VkDfCitlyz/P+xGgZW9YYw5GnDOsEgVHeU.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.1.1.1' (ECDSA) to the list of known hosts.
abc@10.1.1.1's password:
Connected to 10.1.1.1.
sftp>
sftp> get /users/abc/test_file
Fetching /users/abc/test_file to test_file
```

```
/users/abc/test_file               100%  212     0.2KB/s   00:00
sftp> put test_file /users/abc/
Uploading test_file to /users/abc/test_file
test_file                         100%  212     0.2KB/s   00:00
```

```
switch# copy sftp user IPv6-address
```

- Provide the username and IPv6 address of the remote device

```
switch# copy sftp abc a::1
The authenticity of host 'a::1 (a::1)' can't be established.
ECDSA key fingerprint is SHA256:uWeyXm2j6VkDfCitlyz/P+xGgZW9YYw5GnDOsEgVHeU.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'a::1' (ECDSA) to the list of known hosts.
abc@a::1's password:
Connected to a::1.
sftp>
sftp> get /users/abc/test_file
Fetching /users/abc/test_file to test_file
/users/abc/test_file           100%  212     0.2KB/s   00:00
sftp> put test_file /users/abc/
Uploading test_file to /users/abc/test_file
test_file
```

1.8. sFlow

1.8.1. Overview

sFlow is a technology for monitoring traffic in high-speed switched or routed networks. The sFlow monitoring system is comprised of:

- An sFlow Agent that runs on a network device, such as a switch. The agent uses sampling techniques to capture information about the data traffic flowing through the device and forwards this information to an sFlow collector. The OpenSwitch sFlow agent can sample traffic from all physical and bonded interfaces in the system.
- An sFlow Collector that receives monitoring information from sFlow agents. The collector stores this information so that a network administrator can analyze it to understand network data flow patterns.

sFlow datagrams sent to the collector are not encrypted, therefore any sensitive information contained in an sFlow sample is exposed.

1.8.2. Configuring the OpenSwitch sFlow agent

- The agent can communicate with up to three sFlow collectors at the same time.
- Only the default VRF is supported.
- Only in-band collectors are supported.
- Support is provided for the SNMP MIB-2 ifTable.
- Although the CLI allows high sampling rates, the switch may drop samples if it cannot handle the rate of sampled packets.

1.8.3. Default settings

- sFlow is disabled on all interfaces.
- Collector port = 6343.
- Agent interface address type = IPv4.
- Sampling rate = 4096.
- Polling interval = 30 seconds.
- Header size = 128 bytes.
- Max datagram size = 1400 bytes.

1.8.4. Enabling sFlow globally

To enable sFlow on all interfaces on the switch:

```
switch#
switch# configure terminal
switch(config)# sflow enable
```

1.8.5. Disabling sFlow globally

To disable sFlow on all interfaces on the switch:

```
switch#
switch# configure terminal
switch(config)# no sflow enable
```

1.8.6. Configuring collectors

To configure an sFlow collector:

```
switch(config)# sflow collector <ip> [port <port>] [vrf <vrf-name>]
```

- ip is the IPv4 or IPv6 address of the collector.
- port is any valid UDP port that the collector is listening on. Default is 6343.
- vrf-name is the name of a VRFs on the switch. Default is vrf_default.

To remove an sFlow collector:

```
switch(config)# no sflow collector <IP> [port <port>] [vrf <vrf-name>]
```

1.8.7. Configuring the sampling rate

To configure the global sampling rate:

```
switch(config)# sflow sampling <rate>
```

- rate is the approximate number of packets between samples. Default is 4096, which means that approximately every 4096 packet will be sampled. (There is some jitter introduced purposefully into the sample collection.)

To reset the global sampling rate to default:

```
switch(config)# no sflow sampling
```

1.8.8. Configuring the polling interval

To configure the interval at which statistics are send to the collector:

- switch(config)# sflow polling <interval>
- interval is in seconds and the default is 30.

To set the polling interval back to default:

```
switch(config)# no sflow polling
```

1.8.9. Configuring agent interface name and address family

To define sFlow agent interface settings:

```
switch(config)# sflow agent-interface <ifname> [ipv4|ipv6]
```

- ifname is the name of the interface whose IP address will be used as the agent IP in sFlow datagrams. If not specified, system will pick the IP address from one of the interfaces in the switch.
- ipv4 | ipv6 optionally set the address type for the agent interface. By default the address type is IPv4.

1.8.10. Configuring header size

To set the sFlow header size:

```
switch(config)# sflow header-size <size>
```

- size is the the maximum number of bytes to copy and forward from the header of the sampled packet. Default is 128 bytes.

To set the sFlow header size back to default:

```
switch(config)# no sflow header-size
```

1.8.11. Configuring max datagram size

To set the sFlow max datagram size:

- switch(config)# sflow max-datagram-size <size>
- size is the maximum number of bytes that will be sent in one sFlow UDP datagram. Default is 1400 bytes.

To set the sFlow max datagram size back to default:

```
switch(config)# no sflow max-datagram-size
```

1.8.12. Enabling/disabling sFlow per interface

sFlow can be enbled/disabled individually on each interface.

To enable sFlow on a specific interface:

```
switch(config)# interface <interface-name>
switch(config-if)# sflow enable
```

To disable sFlow on a specific interface:

```
switch(config)# interface <interface-name>
switch(config-if)# no sflow enable
```

1.8.13. Viewing sFlow configuration

To view global sFlow configuration settings:

```
switch# show sflow
```

To view sFlow settings for a specific interface:

```
switch# show sflow <interface-name>
```

1.9. Remote Syslog Logging Configuration

1.9.1. Overview

The remote syslog feature enables the switch to forward syslog messages to the remote syslog server.

1.9.1.1. How to use the feature

Setting up the basic configuration

Syntax	logging <IPv4-address> | <IPv6-address> | <hostname> [udp [<port>] | tcp [<port>]] [severity <level>]

Example:

```
switch(config)#logging 10.0.10.9 tcp 4242 severity err
```

Verifying the configuration

- Run the remote syslog server at the specified address, transport, and port.
- Run show running configuration in the switch to verify the configuration.
- Check to ensure that the remote syslog server is receiving the messages as configured.

1.9.2. Troubleshooting the configuration

Condition

The syslog messages are not received on the remote syslog server.

Cause

The following are the possible causes for this problem:

- Remote syslog server is not running.
- Remote syslog server is not reachable from the switch.
- Remote syslog server has an improper configuration.

Remedy

- Confirm that the remote syslog server is running.
- Test the reachability of the remote syslog server from the switch. You can use the ping command to check this.
- Make sure that the remote syslog server is configured to receive remote syslog messages.

1.10. Access Control List (ACL)

1.10.1. Overview

ACLs can be used to help improve network performance and restrict network usage by creating policies to eliminate unwanted IP traffic by filtering packets where they enter the switch on layer 2 and layer 3 interfaces.

An access control list (ACL) is an ordered list of one or more access control list entries (ACEs) prioritized by sequence number. An incoming packet is matched sequentially against each entry in an ACL. When a match is made, the action of that ACE is taken and the packet is not compared against any other ACEs in the list.

For ACL filtering to take effect, configure an ACL and then assign it in the inbound direction on either a layer 2 or layer 3 interface.

Every ACL is configured with a deny any any any entry as the last entry in the list. This entry is known as the implicit deny entry, and applies to all IP traffic that does not match any sequentially higher (numerically lower) entry in the list. The implicit deny entry will not filter traffic of a type different from the ACL. For example, an IP ACL applies to IPv4 traffic only, it will not filter IPv6 traffic. An ACL with no user configured entries that is applied to an interface, is programmed in hardware with the implicit deny entry.

ACLs are supported on split interfaces. ACLs are not supported on sub-interfaces.

1.10.2. Setting up the basic configuration

Prior to applying an ACL to an interface, verify that traffic is flowing as expected through the device.

Create an ACL:

```
switch(config)# access-list ip testACL
```

Add one or more entries to the list:

```
switch(config-acl)# 10 permit udp any 172.16.1.0/24
switch(config-acl)# 20 permit tcp 172.16.2.0/16 gt 1023 any
switch(config-acl)# 30 deny any any any count
switch(config-acl)# exit
```

Apply the ACL to an interface:

```
switch(config)# interface 1
switch(config-if)# apply access-list ip testACL in
switch(config-if)# exit
```

Delete an ACL

```
switch(config)# no access-list ip testACL
```

If an ACL is deleted while applied to any interfaces, the ACL automatically is unapplied from all interfaces.

1.10.3. Verifying the configuration

Use the following show commands to display the ACL configuration:

Syntax show access-list [interface <id> [in]] [ip] [<acl-name>] [commands] [configuration]

The following refers to the above basic configuration example.

Show the ACL applied to interface 1:

```
switch# show access-list interface 1 in
Direction
Type       Name
 Sequence Comment
      Action                          L3 Protocol
      Source IP Address               Source L4 Port(s)
      Destination IP Address          Destination L4 Port(s)
      Additional Parameters
-------------------------------------------------------------------------------
Inbound
IPv4       testACL
   10 permit                          udp
      any
      172.16.1.0/255.255.255.0
   20 permit                          tcp
      172.16.2.0/255.255.0.0           >  1023
      any
   30 deny                            any
      any
      any
      Hit-counts: enabled
-------------------------------------------------------------------------------
```

Show all active ACLs configured on the system:

```
switch# show access-list
Type       Name
 Sequence Comment
      Action                          L3 Protocol
      Source IP Address               Source L4 Port(s)
      Destination IP Address          Destination L4 Port(s)
      Additional Parameters
-------------------------------------------------------------------------------
IPv4       100
   10 permit                          tcp
      1.1.1.1/255.255.255.0
      2.2.2.2/255.255.255.0            >  1024
   20 permit                          udp
      1.1.1.1/255.255.255.0
      2.2.2.2/255.255.255.0            >  1024
-------------------------------------------------------------------------------
IPv4       testACL
   10 permit                          udp
      any
      172.16.1.0/255.255.255.0
   20 permit                          tcp
      172.16.2.0/255.255.0.0           >  1023
      any
   30 deny                            any
      any
      any
      Hit-counts: enabled
-------------------------------------------------------------------------------
```

Show all ACLs configured by the user:

```
switch# show access-list configuration
Type       Name
 Sequence Comment
      Action                          L3 Protocol
      Source IP Address               Source L4 Port(s)
      Destination IP Address          Destination L4 Port(s)
      Additional Parameters
-------------------------------------------------------------------------------
IPv4       100
   10 permit                          tcp
      1.1.1.1/255.255.255.0
      2.2.2.2/255.255.255.0            >  1024
   20 permit                          udp
      1.1.1.1/255.255.255.0
      2.2.2.2/255.255.255.0            >  1024
-------------------------------------------------------------------------------
IPv4       testACL
   10 permit                          udp
      any
      172.16.1.0/255.255.255.0
   20 permit                          tcp
      172.16.2.0/255.255.0.0           >  1023
      any
   30 deny                            any
      any
      any
      Hit-counts: enabled
-------------------------------------------------------------------------------
```

Show the ACLs in command line format:

```
switch# show access-list commands
```

```
access-list ip 100
   10 permit tcp 1.1.1.1/255.255.255.0 2.2.2.2/255.255.255.0 gt 1024
   20 permit udp 1.1.1.1/255.255.255.0 2.2.2.2/255.255.255.0 gt 1024
access-list ip testACL
   10 permit udp any 172.16.1.0/255.255.255.0
   20 permit tcp 172.16.2.0/255.255.0.0 gt 1023 any
   30 deny any any any count
interface 1
   apply access-list ip testACL in
```

1.10.4. Active configuration versus user-specified configuration

The output from the show access-list command displays the active configuration of the product. The active configuration displays the ACLs that have been configured and accepted by the system. In the case of applied ACLs, the active configuration displays the interfaces on which the ACLs have successfully been programmed in hardware.

The output from the show access-list command with the configuration option, displays the ACLs that have been configured by the user. The output of this command may not be the same as what was programmed in hardware or what is active on the product. Unsupported command parameters may have been configured, unsupported applications may have been specified, or an application of an ACL may have been unsuccessful due to a lack of hardware resources. To determine if there is a discrepancy between what was configured and what is active, run either the show access-list commands command or the show access-list commands configuration command. If the active ACLs and configured ACLs are not the same, a warning message is displayed.

Warning: user-specified access-list apply does not match active configuration

If the warning message is displayed, additional changes may be made until the error message is no longer displayed when show access-list commands or show access-list commands configuration is entered, or run the reset access-list command. The reset access-list command changes the user-specified configuration to match the active configuration.

```
reset access-list {all | ip <acl-name>}
```

Example:

```
switch(config)# access-list ip testACL
switch# show access-list commands configuration
```

Change the source L4 port operator of entry 20 to be neq, which is unsupported by hardware:

```
switch(config-acl)# 20 permit tcp 172.16.2.0/16 neq 1023 any
```

Display the user-specified configuration:

```
switch# show access-list commands configuration
access-list ip 100
   10 permit tcp 1.1.1.1/255.255.255.0 2.2.2.2/255.255.255.0 gt 1024
   20 permit udp 1.1.1.1/255.255.255.0 2.2.2.2/255.255.255.0 gt 1024
access-list ip testACL
   10 permit udp any 172.16.1.0/255.255.255.0
   20 permit tcp 172.16.2.0/255.255.0.0 neq 1023 any /* neq is an unsupported parame
   30 deny any any any count
interface 1
   apply access-list ip testACL in
% Warning: user-specified access-list apply does not match active configuration
```

Reset the user-specified configuration to the active configuration:

```
switch(config)# reset access-list all
```

Display the updated user-configuration:

```
switch# show access-list commands configuration
access-list ip 100
   10 permit tcp 1.1.1.1/255.255.255.0 2.2.2.2/255.255.255.0 gt 1024
   20 permit udp 1.1.1.1/255.255.255.0 2.2.2.2/255.255.255.0 gt 1024
access-list ip testACL
   10 permit udp any 172.16.1.0/255.255.255.0
   20 permit tcp 172.16.2.0/255.255.0.0 gt 1023 any
   30 deny any any any count
interface 1
   apply access-list ip testACL in
```

1.10.5. Displaying ACL hitcounts

Hitcounts are available for ACEs that are created with the count keyword specified, as in entry 30 in the example ACL testACL. The ACL must be applied to an interface and actively configured for the hitcounts to be valid.

```
switch# show access-list hitcounts ip testACL interface 1 in
Statistics for ACL testACL (ipv4):
Interface 1 (in):
          Hit Count  Configuration
                  -  10 permit udp any 172.16.1.0/255.255.255.0
                  -  20 permit tcp 172.16.2.0/255.255.0.0 gt 1023 any
               2045  30 deny any any any count
```

In the above output, only entry 30 displays a numerical hitcount. Entry 30 is the only ACE that contains the count keyword.

1.10.6. ACL Logging

ACL logging is a useful tool for troubleshooting ACLs. It can identify traffic in the following scenarios:

- Traffic is expected to be permitted and is blocked.
- Unexpected traffic in the network.

When the log keyword is specified for an ACE, packets that match this entry are copied to the switch CPU for processing by the ACL logging feature. The first packet that matches any log-enabled ACE in an ACL are logged to the configured system log. The reception of this packet also starts the ACL logging timer. Subsequent packets that match log-enabled ACEs are not logged to the system log for the duration of the ACL logging timer. When the ACL logging timer expires, a list of all the ACLs with logging enabled ACEs and their corresponding hitcounts, are logged to the system log. The next packet that matches a log-enabled ACE is logged, repeating the process of starting the ACL logging timer. The ACL logging feature displays the ACL hitcounts stored in the database, which are read from hardware every five seconds.

Setting the log keyword in an ACE automatically sets the count keyword in the ACE, enabling statistics for the purpose of reporting hitcounts when the logging timer expires.

Configure an ACL entry for logging:

```
switch(config)# access-list ip testACL
switch(config-acl)# 30 deny any any any log
```

Display logging messages through the vtysh shell:

```
switch# show vlog daemon ops-switchd
```

Sample output:

```
ops-switchd              |ovs|00833|ops_cls_acl_log|INFO|List testACL,
seq#30 denied udp 172.16.2.15(1564) -> 17.77.115.80(1780) on vlan 1, port 1,
direction in
```

Output after the ACL logging timer expires:

```
ops-switchd              |ovs|00838|ops_cls_acl_log|INFO|testACL on 1 (in):
2544008  30 deny any any any log
```

Display logging messages through the Linux shell:

```
root@switch:~# tail -f /var/log/messages | grep ops_cls_acl_log
```

```
2016-06-05T00:00:03.432+00:00 ops-switchd: ovs|05237|ops_cls_acl_log|INFO|List
testACL, seq#30 denied 172.16.2.15(1564) -> 17.77.115.80(1780) on vlan 1, port 3,
direction in
2016-06-05T00:00:33.507+00:00 ops-switchd: ovs|05238|ops_cls_acl_log|INFO|test on 1
(in):      2540076  30 deny any any any log
```

1.10.7. Configure the ACL logging timer

```
access-list log-timer 30
```

The default and maximum duration of the ACL logging timer is 300 seconds. The minimum duration is 30 seconds.

1.10.8. Displaying hardware resource usage

ACLs consume resources from a limited hardware pool. An ACL does not consume hardware resources until it is applied to an interface. More ACLs may be created than can be applied in hardware. Use the following command to display the consumption of the hardware resources:

```
switch# diag-dump hw-resource basic
```

Depending on the hardware implementation of ACLs in the product, the command displays something similar to the following:

```
=========================================================================
[Start] Feature hw-resource Time : Sat Jun  4 12:07:03 2016
=========================================================================
-------------------------------------------------------------------------
[Start] Daemon ops-switchd
-------------------------------------------------------------------------
```

```
Hardware resource usage:
Ingress:
              |Rules |Rules   |Group |Group |Counters |Counters |Meters |Meters
Feature       |Used  |Maximum |ID    |Used  |Used     |Maximum  |Used   |Maximum
--------------|------|--------|------|------|---------|-------- |-------|-------
ospf          |     2|    2048|     1|     1|        2|     2048|      0|   4096
copp          |    20|    1280|     2|     1|        0|     1792|      0|   4096
aclv4         |     4|     512|     4|     1|        0|      512|      0|   4096
l3intf        |     8|    1792|     5|     1|        8|     1792|      0|   4096
```

```
Egress:
              |Rules |Rules   |Group |Group |Counters |Counters |Meters |Meters
Feature       |Used  |Maximum |ID    |Used  |Used     |Maximum  |Used   |Maximum
--------------|------|--------|------|------|---------|-------- |-------|-------
copp          |  20  |     512|     1|     1|       40 |    1024 |     40|    768
```

1.10.9. Troubleshooting the configuration

IP traffic that was not explicitly denied is blocked

Cause

The implicit deny entry is blocking the traffic.

Remedy

Add an ACE to explicitly permit the traffic.

An empty ACL is applied to an interface and all IP traffic is blocked

Cause

The implicit deny entry is blocking the traffic.

Remedy

Add an ACE to explicitly permit the traffic.

An ACL is applied and is not blocking traffic

Cause

The ACL has been configured but is not active.

Remedy

Run the command show access-list commands. If a warning message is displayed, issue the reset access-list command. Re-run show access-list commands to verify the warning message is no longer displayed.

Permitted traffic is denied or denied traffic is permitted

Cause

Misconfigured ACL.

Remedy

Enable the count keyword on the troublesome ACEs. While sending traffic, run the ACL hitcount show command and monitor the counts of the ACEs.

1.11. Show Events

1.11.1. Overview

The show events command is used to display events for all of the supported features. This command is useful to generate switch event logs for administrators, developers, support staff, and lab staff. Problem events and solutions are also easily obtainable using the CLI.

1.11.2. How to use the CLI

To access events for supported features or daemons in the switch, run the CLI **show events** command.

1.11.3. Setting up the basic configuration

The show events command infrastructure loads its configuration from the "show event configuration yaml" file located in (/etc/openswitch/supportability/ops_events.yaml).

This file contains the default configuration for the show events command.

1.11.4. Verifying the configuration

Execute the CLI **show events** command and verify that the features are configured.

1.11.5. Log filter options

The show events command provides log filter options to show only the logs of interest. The following filters are supported:

- event-id
- severity
- category

These filter keywords can be used along with the show events command to filter logs accordingly.

Examples

To filter according to event ID 1002:

```
show events event-id 1002
```

To filter according to severity level of emergency:

```
show events severity emer
```

The following are the severity keywords supported in the CLI:

- emer

- alert
- crit
- error
- warn
- notice
- info
- debug

To filter according to interested log category:

```
show events category LLDP
```

A combination of all of these filters can also be used:

```
show events event-id 1003 category LLDP severity emer
```

1.11.6. Reverse list option

The show events output usually displays from oldest to latest in order. You can make use of the reverse keyword to list logs from most recent to oldest. This option can be used along with any of the log filters as well.

For example:

```
show events reverse show events event-id 1003 category LLDP severity emer reverse
```

1.11.7. Troubleshooting the configuration

Configuration file is missing in its path

If the error ops_events.yaml configuration file is missing in its path appears, ensure that the ops_events.yaml file is present in the (/etc/openswitch/supportability/ops_events.yaml) path.

File is not properly configured

If the error ops_events.yaml configuration file is wrongly configured appears, use the yaml tools to confirm that the configuration file (ops_events.yaml) is valid.

1.12. Ping Utility

1.12.1. Overview

The ping (Packet InterNet Groper) command is a very common method for troubleshooting the accessibility of devices. It uses Internet Control Message Protocol (ICMP) echo requests and ICMP echo replies to determine if another device is alive. It also measures the amount of time it takes to receive a reply from the specified destination. The ping command is mostly used to verify connectivity between your switch and a host or port. The reply packet tells you if the host received the ping and the amount of time it took to return the packet.

Syntax ping <ipv4-address | hostname> [repetitions <1-10000>] [timeout <1-60>] [interval <1-60>] [datagram-size <100-65399>] [data-fill <WORD>][ip-option (include-timestamp |include-timestamp-and-address |record-route)][tos <0-255>]

Syntax ping6 <ipv6-address | hostname> [repetitions <1-10000>] [interval <1-60>] [datagram-size <100-65468>] [data-fill <WORD>]

Explanation of parameters

- Ipv4-address - Target IPv4 address of the destination node being pinged.
- Hostname - Hostname of the destination node being pinged.
- Repetitions <1-10000> - Number of ping packets sent to the destination address. The default value is 5.
- Timeout <1-60> - Timeout interval in seconds, the ECHO REPLY must be received before this time interval expires for the Ping to be successful. The default value is 2 seconds.
- Interval <1-60> - Interval seconds between sending each packet. The default value is 1 second.
- Datagram-size - Size of packet sent to the destination. The default value is 100 bytes.
- Data-fill - Hexadecimal pattern to be filled in the packet. A Maximum of 16 *pad* bytes can be specified to fill out the icmp packet.
- Ip-options - This prompt offers more selection of any one option from the list below.
- Include-timestamp - Timestamp option is used to measure roundtrip time to particular hosts.
- Include-timestamp-and-address - Displays roundtrip time to particular hosts as well as address.
- Record-route - Displays the addresses of the hops the packet goes through.
- TOS <0-255> - Specifies the Type of Service (TOS). The requested TOS is placed in each probe. It is the Internet service quality selection.

Ping 0.0.0.0 and ping6 0::0 issues ping and ping6 to localhost. This is default Linux behavior.

1.12.2. Ping IPv4-address

Success case

Send an IP ping request to the device that has IP address 9.0.0.1.

```
switch# ping 9.0.0.1
PING 9.0.0.1 (9.0.0.1) 100(128) bytes of data.
108 bytes from 9.0.0.1: icmp_seq=1 ttl=64 time=0.035 ms
108 bytes from 9.0.0.1: icmp_seq=2 ttl=64 time=0.034 ms
108 bytes from 9.0.0.1: icmp_seq=3 ttl=64 time=0.034 ms
108 bytes from 9.0.0.1: icmp_seq=4 ttl=64 time=0.034 ms
108 bytes from 9.0.0.1: icmp_seq=5 ttl=64 time=0.033 ms
```

```
--- 9.0.0.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3999ms
rtt min/avg/max/mdev = 0.033/0.034/0.035/0.000 ms
```

Send an IP ping request to IP address 0.0.0.0.

```
switch# ping 0.0.0.0
PING 0.0.0.0 (127.0.0.1) 100(128) bytes of data.
108 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.061 ms
108 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.065 ms
108 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.061 ms
108 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.041 ms
108 bytes from 127.0.0.1: icmp_seq=5 ttl=64 time=0.036 ms
```

```
--- 0.0.0.0 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4002ms
rtt min/avg/max/mdev = 0.036/0.052/0.065/0.015 ms
```

Failure case

Network unreachable

```
switch# ping 1.1.1.1
connect: Network is unreachable
```

Destination host unreachable

```
switch# ping 9.0.0.1
PING 9.0.0.1 (9.0.0.1) 100(128) bytes of data.
From 9.0.0.2 icmp_seq=1 Destination Host Unreachable
From 9.0.0.2 icmp_seq=2 Destination Host Unreachable
From 9.0.0.2 icmp_seq=3 Destination Host Unreachable
From 9.0.0.2 icmp_seq=4 Destination Host Unreachable
```

```
--- 9.0.0.1 ping statistics ---
5 packets transmitted, 0 received, +4 errors, 100% packet loss, time 4002ms
pipe 4
```

1.12.3. Hostname

```
ping hostname
```

Domain name of the host to ping.

1.13. Repetitions

```
ping <ipv4-address | hostname> repetitions <1-10000>
```

Number of packets to send <1-10000>. Range: < 1 to 10000 >

1.13.1. Timeout

```
ping <ipv4-address | hostname> timeout <1-60>
```

Ping timeout in seconds <1-60>. Range: <1 to 60>

1.13.2. Interval

```
ping <ipv4-address | hostname> interval <1-60>
```

Seconds between sending each packet <1-60>. Range: <1 to 60>

1.13.3. Data-fill

```
ping <ipv4-address | hostname> data-fill WORD
```

Ping data fill string, example *ab* A Maximum of 16 *pad* bytes can be specified to fill out in the icmp packet.

1.13.4. Datagram-size

```
ping <ipv4-address | hostname> datagram-size <100-65399>
```

Range: <100 to 65399>

1.13.5. Ping IPv6-address

Success case

Send an IPv6 Ping request to the device that has IPv6 address 2020::2

```
switch# ping6 2020::2
PING 2020::2(2020::2) 100 data bytes
108 bytes from 2020::2: icmp_seq=1 ttl=64 time=0.386 ms
108 bytes from 2020::2: icmp_seq=2 ttl=64 time=0.235 ms
108 bytes from 2020::2: icmp_seq=3 ttl=64 time=0.249 ms
108 bytes from 2020::2: icmp_seq=4 ttl=64 time=0.240 ms
108 bytes from 2020::2: icmp_seq=5 ttl=64 time=0.252 ms
```

~~~
--- 2020::2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4000ms
rtt min/avg/max/mdev = 0.235/0.272/0.386/0.059 ms
```
~~~

Send an IPv6 Ping request to IPv6 address 0::0

```
switch# ping6 0::0
PING 0::0(::) 100 data bytes
108 bytes from ::1: icmp_seq=1 ttl=64 time=0.036 ms
108 bytes from ::1: icmp_seq=2 ttl=64 time=0.064 ms
108 bytes from ::1: icmp_seq=3 ttl=64 time=0.065 ms
108 bytes from ::1: icmp_seq=4 ttl=64 time=0.060 ms
108 bytes from ::1: icmp_seq=5 ttl=64 time=0.061 ms
--- 0::0 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4000ms
rtt min/avg/max/mdev = 0.036/0.057/0.065/0.011 ms
```

Failure case

Network unreachable

```
switch# ping6 3030::1
connect: Network is unreachable
```

Destination host unreachable

```
switch# ping6 2020::1
PING 2020::1(2020::1) 100 data bytes
From 2020::2 icmp_seq=1 Destination unreachable: Address unreachable
From 2020::2 icmp_seq=2 Destination unreachable: Address unreachable
From 2020::2 icmp_seq=3 Destination unreachable: Address unreachable
From 2020::2 icmp_seq=4 Destination unreachable: Address unreachable
From 2020::2 icmp_seq=5 Destination unreachable: Address unreachable
--- 2020::1 ping statistics ---
5 packets transmitted, 0 received, +5 errors, 100% packet loss, time 4000ms
```

1.13.6. Hostname

```
ping6 hostname
```

Domain name of the host to ping.

1.13.7. Repetitions

```
ping6 <ipv6-address | hostname> repetitions <1-10000>
```

Number of packets to send <1-10000>.

Range: <1 to 10000>

1.13.8. Interval

```
ping6 <ipv6-address | hostname> interval <1-60>
```

Seconds between sending each packet <1-60>.

Range: <1 to 60>

1.13.9. Data-fill

```
ping6 <ipv6-address | hostname> data-fill WORD
```

Ping data fill string, example *ab*

A Maximum of 16 *pad* bytes can be specified to fill out in the icmp packet.

1.13.10. Datagram-size

```
ping6 <ipv6-address | hostname> datagram-size <100-65468>
```

Range: <100 to 65468>

1.14. Traceroute Utility

1.14.1. Overview

Traceroute is a computer network diagnostic tool for displaying the route (path), and measuring transit delays of packets across an Internet Protocol (IP) network. It sends a sequence of User Datagram Protocol (UDP) packets addressed to a destination host. The time-to-live (TTL) value, also known as hop limit, is used in determining the intermediate routers being traversed towards the destination.

Syntax traceroute <IP-ADDR | hostname > [dstport <1-34000>] [maxttl <1-255>] [minttl <1-255>] [probes <1-5>] [timeout <1-60>] [ip-option loosesourceroute <IP-ADDR>]

Syntax traceroute6 <IP-ADDR | hostname > [dstport <1-34000>] [maxttl <1-255>] [probes <1-5>] [timeout <1-60>]

Explanation of parameters

- IP-ADDR - Network IP address of the device to which to send traceroute.
- Hostname - Domain name of the device to which to send traceroute.
- Dstport <1-34000> -Destination port For UDP tracing. The default value is 33434.
- Maxttl <1-255> - Maximum number of hops used in outgoing probe packets. The default value is 30.
- Minttl <1-255> - Minimum number of hops used in outgoing probe packets. The default value is 1.
- Probes <1-5> - Number of probe queries to send out for each hop. The default value is 3.
- Timeout <1-60> - Time (in seconds) to wait for a response to a probe. The default value is 3 seconds.
- Ip-option - Tells traceroute to add an IP source routing option to the outgoing packet.
- Loosesourceroute - Tells the network to route the packet through the specified gateway.

1.14.2. Traceroute IP-address

Send IP traceroute UDP packets to the device that has IP address 10.0.10.1:

```
switch# traceroute 10.0.10.1
traceroute to 10.0.10.1 (10.0.10.1) , 1 hops min, 30 hops max, 3 sec.
timeout, 3 probes
1   10.0.40.2  0.002ms  0.002ms  0.001ms
2   10.0.30.1  0.002ms  0.001ms  0.001ms
3   10.0.10.1  0.001ms  0.002ms  0.002ms
```

Send IP traceroute UDP packets to the device that has IP address 0.0.0.0:

```
switch# traceroute 0.0.0.0
```

```
traceroute to 0.0.0.0 (0.0.0.0), 1 hops min, 30 hops max, 3 sec.
timeout, 3 probes
1   127.0.0.1  0.015ms  0.003ms  0.002ms
```

Failure case

Network unreachable

```
switch# traceroute 10.0.0.1
traceroute to 10.0.0.1 (10.0.0.1), 1 hops min, 30 hops max, 3 sec.
timeout, 3 probes
1  traceroute: sendto: Network is unreachable
```

Destination host unreachable

```
switch# traceroute 9.0.0.6
traceroute to 9.0.0.6 (9.0.0.6), 1 hops min, 30 hops max, 3 sec.
timeout, 3 probes
1 9.0.0.2  2 ms !H  3 ms !H  2 ms !H
```

1.14.3. Hostname

```
traceroute hostname
```

Domain name of the host to traceroute.

1.14.4. Destination port

```
traceroute  <IP | Hostname> dstport <1-34000>.
```

Destination port number.

Range: <1 to 34000>

1.14.5. Maximum TTL

```
traceroute  <IP | Hostname> maxttl <1-255>
```

Maximum number of hops used in outgoing probe packets.

Range: <1 to 255>

1.14.6. Minimum TTL

```
traceroute <IP | Hostname> minttl <1-255>
```

Minimum number of hops used in outgoing probe packets.

Range: <1 to 255>

1.14.7. Probes

```
traceroute  <IP | Hostname> probes <1-5>
```

Number of probe queries to send out for each hop <1-5>.

Range: < 1 to 5 >

1.14.8. Timeout

```
traceroute  <IP | Hostname> timeout <1-60>
```

Time (in seconds) to wait for a response to a probe <1-60>.

Range: < 1 to 60 >

1.14.9. Ip-option loose source route

```
traceroute <IP | Hostname> ip-option loosesourceroute <IP-ADDR>
```

Loose source route defines the default gateway to the destination.

1.14.10. Traceroute6 IPv6-address

Send IPv6 traceroute UDP packets to the device that has IPv6 address 0:0::0:1 :

```
switch# traceroute6 0:0::0:1
traceroute to 0:0::0:1 (::1) from ::1, 30 hops max, 3 sec.
timeout, 3 probes, 24 byte packets
1  localhost (::1)  0.117 ms  0.032 ms  0.021 ms
```

Send IP traceroute UDP packets to the device that has IP address ::0:

```
switch# traceroute6 ::0
traceroute to ::0 (::) from ::1, 30 hops max, 3 sec.
timeout, 3 probes, 24 byte packets
1  localhost (::1)  0.144 ms  0.03 ms  0.016 ms
```

Failure case

Network unreachable

```
switch# traceroute6 2004::2
connect: Network is unreachable
```

Destination host unreachable

```
switch# traceroute6 2001::2
traceroute to 2001::2 (2001::2) from 2001::1, 30 hops max, 3 sec. timeout, 3 probes,
1  switch (2001::1)  2998.93 ms !H  2999.14 ms !H  2999.79 ms !H
```

1.14.11. Hostname

```
traceroute6 hostname
```

Domain name of the host to traceroute6.

1.14.12. Destination port

```
traceroute6 <IPv6 | Hostname> dstport <1-34000>
```

Destination port number.

Range: <1 to 34000>

1.14.13. Maximum TTL

```
traceroute6 <IPv6 | Hostname> maxttl <1-255>
```

Maximum number of hops used in outgoing probe packets.

Range: <1 to 255>

1.14.14. Probes

```
traceroute6 <IPv6 | Hostname> probes <1-5>
```

Number of probe queries to send out for each hop <1-5>.

Range: < 1 to 5 >

1.14.15. Timeout

```
traceroute6 <IPv6 | Hostname> timeout <1-60>
```

Time (in seconds) to wait for a response to a probe <1-60>.

Range: < 1 to 60 >

1.15. Diagnostic Dump Commands

1.15.1. Overview

The diagnostic dump command lets you display or save diagnostic information about an OpenSwitch feature.

1.15.2. Listing features that support diagnostic dump

Run following command to see all features that support diagnostic information:

```
diag-dump list
```

Example

```
switch# diag-dump list
Diagnostic Dump Supported Features List
-------------------------------------------------------------------------
Feature                                  Description
-------------------------------------------------------------------------
lldp                                     Link Layer Discovery
lacp                                     Link Aggregation Con
fand                                     System Fan
routing                                  Routing protocols
ospfv2                                   Open Shortest Path F
bgp                                      Border Gateway Proto
sub-interface                            sub-interface
loopback                                 Loopback interface
```

1.15.3. Viewing basic diagnostic information for a feature

Run the command:

```
diag-dump <feature-name> basic
```

Example

```
switch# diag-dump lldp basic
=========================================================================
[Start] Feature lldp Time : Thu Apr 14 02:54:18 2016
=========================================================================
-------------------------------------------------------------------------
[Start] Daemon ops-lldpd
-------------------------------------------------------------------------
LLDP : DISABLED
   intf name   |   OVSDB interface     |   LLDPD Interface     |    LLDP Status
====================================================================================
bridge_normal   |    Yes                |    Yes                |    rxtx
51-3            |    Yes                |No              |
53              |    Yes                |No             |
49-4            |    Yes                |No             |
51-1            |    Yes                |No             |
35              |    Yes                |No             |
6               |    Yes                |No             |
39              |    Yes                |No             |
4               |    Yes                |No             |
....
```

```
-------------------------------------------------------------------------
[End] Daemon ops-lldpd
-------------------------------------------------------------------------
=========================================================================
[End] Feature lldp
=========================================================================
Diagnostic dump captured for feature lldp
```

1.15.4. Saving basic diagnostic information to a file

Run the command:

```
diag-dump <feature-name> basic <filename>
```

1.15.5. Troubleshooting

Condition

The diag-dump command displays the message: Failed to capture diagnostic information

Causes

- The configuration file (/etc/openswitch/supportability/ops_featuremapping.yaml) is missing.
- You may not have read permission for the configuration file.
- The content of the configuration file is incorrect.

Remedy

- Ensure that the configuration file (/etc/openswitch/supportability/ops_featuremapping.yaml) exists.
- Ensure that the you have read permission for the configuration file.
- Verify that the content of the configuration file is correct using the yaml lint tool.
- Verify that the structure of the configuration file is valid.

1.16. Core Dump CLI Guide

1.16.1. Overview

This command copies a daemon or a kernel corefile to the tftp or sftp server.

1.16.2. How to use the feature

Use the following procedure to copy the daemon core to a TFTP server:

- Execute the command sequence copy core-dump <DAEMONNAME> instance-id <INSTANCE ID> tftp <TFTP SERVER IPV4 ADDRESS /HOST NAME> [FILENAME]
- Execute the command sequence copy core-dump <DAEMONNAME> tftp <TFTP SERVER IPV4 ADDRESS / HOSTNAME >

Use the following procedure to copy the daemon core to a sftp server:

- Execute the command sequence copy core-dump <DAEMONNAME> instance-id <INSTANCE ID> sftp <USERNAME> <SFTP SERVER IPV4/HOST NAME> [FILENAME].
- Execute the command sequence copy core-dump < DAEMON NAME> sftp <USERNAME> <SFTP SERVER ADDRESS> [DESTINATION FILE NAME].

Use the following procedure to copy the kernel core:

- Execute the command sequence copy core-dump kernel tftp <TFTP SERVER IPV4 ADDRESS / HOST NAME > [DESTINATION FILE NAME].
- Execute the command sequence copy core-dump kernel sftp <USERNAME> <SFTP SERVER IP> [DESTINATION FILE NAME].

Examples

```
switch# show  core-dump
====================================================================================
Daemon Name         | Instance ID | Crash Reason                  | Timestamp
====================================================================================
ops-vland             439           Aborted                        2016-04-26 18:05:
ops-vland             410           Aborted                        2016-04-26 18:08:
====================================================================================
Total number of core dumps : 2
====================================================================================
```

```
switch#
switch# copy core-dump ops-vland instance-id 439 tftp 10.0.12.161 ops-vland.xz
copying ...
Sent 109188 bytes in 0.1 seconds
switch#
```

```
switch# copy core-dump ops-vland tftp 10.0.12.161
copying ...
Sent 109188 bytes in 0.1 seconds
copying ...
```

```
Sent 109044 bytes in 0.0 seconds
switch#
```

```
switch# copy core-dump kernel tftp   10.0.12.161
copying ...
Sent 30955484 bytes in 19.6 seconds
switch#
```

```
switch#copy core-dump ops-vland instance-id 410 sftp naiksat 10.0.12.161
ops-vland.xz
copying ...
naiksat@10.0.12.161's password:
Connected to 10.0.12.161.
sftp> put /var/diagnostics/coredump/core.ops-vland.0.a6f6c4b58aa5467ba57d7f
9492afa10f.410.1461694139000000.xz ops-vland.xz
Uploading /var/diagnostics/coredump/core.ops-vland.0.a6f6c4b58aa5467ba57d7f
9492afa10f.410.1461694139000000.xz to /users/naiksat/ops-vland.xz
/var/diagnostics/coredump/core.ops-vland.0.a6 100%  106KB 106.5KB/s   00:00
switch#
```

```
switch# copy core-dump ops-switchd sftp naiksat 10.0.12.161
copying ...
The authenticity of host '10.0.12.161 (10.0.12.161)' can't be established.
ECDSA key fingerprint is SHA256:uWeyXm2j6VkDfCitlyz/P+xGgZW9YYw5GnDOsEgVHeU.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.12.161' (ECDSA) to the list of known hosts.
naiksat@10.0.12.161's password:
Connected to 10.0.12.161.
sftp> put /var/diagnostics/coredump/core.ops-switchd.0.1bddabce7ee5468884cc01
2b1d7c2ab0.361.1461694266000000.xz core.ops-switchd.0.1bddabce7ee5468884cc
012b1d7c2ab0.361.1461694266000000.xz
Uploading /var/diagnostics/coredump/core.ops-switchd.0.1bddabce7ee5468884cc0
12b1d7c2ab0.361.1461694266000000.xz to /users/naiksat/core.ops-switchd.0.1bd
dabce7ee5468884cc012b1d7c2ab0.361.1461694266000000.xz
/var/diagnostics/coredump/core.ops-switchd.0. 100% 4531KB   4.4MB/s   00:00
switch#
```

```
switch# copy core-dump kernel sftp naiksat 10.0.12.161 kernelcore.tar.gz
copying ...
naiksat@10.0.12.161's password:
Connected to 10.0.12.161.
sftp> put /var/diagnostics/coredump/kernel-core/vmcore.20160426.180420.tar.gz
kernelcore.tar.gz
Uploading /var/diagnostics/coredump/kernel-core/vmcore.20160426.180420.tar.gz to
/users/naiksat/kernelcore.tar.gz
/var/diagnostics/coredump/kernel-core/vmcore. 100%   44MB  43.5MB/s   00:01
switch#
```

1.16.3. show core dump

This command lists all the core dumps in the switch. Each entry in the listing displays the daemon name that crashed and the timestamp of the crash event.

Examples

```
switch# show core-dump
===========================================================
TimeStamp           | Daemon Name
===========================================================
2016-10-09 09:08:22   ops-lldpd
2015-09-12 10:34:56   kernel
===========================================================
Total number of core dumps : 1
===========================================================
```

If there are no core dumps to present, the following information appears:

```
switch# show core-dump
No core dumps are present
```

1.17. NTP

1.17.1. Overview

The NTP Client functionality is supported on the switch. The switch synchronizes its time with a NTP server using the NTP protocol over a (WAN or LAN) UDP network.

1.17.2. Prerequisites

- An NTP server (either local or remote) is needed with which the switch can synchronize its time.
- OpenSwitch needs to have management interface UP and enabled.

1.17.3. Limitations

- Only the NTP Client functionality alone is supported.
- A maximum of eight servers can be configured from which to synchronize.
- Currently IPv4/IPv6 address is supported. However, multicast, broadcast or loopback addresses are not supported.
- Currently only default VRF is supported.
- REF-ID for the NTPv6 server is compressed from the IPv6 to the IPv4 address.

1.17.4. Defaults

- NTP is enabled by default and cannot be disabled.
- NTP authentication is disabled by default.
- The default NTP version used is 3.

1.17.5. Configuring an NTP client

Configure the terminal to change the CLI context to config context with the following commands:

```
switch# configure terminal
switch(config)#
```

Enabling NTP

NTP is enabled by default and cannot be disabled.

Adding a server

Add a NTP server with the following command:

```
switch(config)# ntp server <FQDN/IPv4/IPv6 address> [key _key-id_]
[version _version-no_] [prefer]
```

- The server name can be a maximum of 57 characters long.
- The IPv4/IPv6 address, if included, needs to be of a valid format.
- Default version is 3. (Only versions 3 or 4 are acceptable values.)

Deleting a server

Delete a previously added NTP server using the following command:

```
switch(config)# no ntp server <FQDN/IPv4/IPv6 address>
```

Enabling NTP authentication

The switch can be configured to authenticate the NTP server to which it synchronizes. The NTP server should already be configured with some authentication keys. One of these keys is used when adding the server on the switch. When NTP authentication is enabled, the switch synchronizes to the NTP server only if the switch has an authentication key specified as a trusted key. Without this, NTP packets are dropped due to an authentication check failure.

```
switch(config)# ntp authentication enable
```

Disabling NTP authentication

Disable NTP authentication using the corresponding no command:

```
switch(config)# no ntp authentication enable
```

Adding an NTP authentication key

When NTP authentication is enabled on the switch, the incoming NTP packets are authenticated using an authentication key. This key needs to be marked as "trusted", and must be the same key previously configured on the NTP server.

```
switch(config)# ntp authentication-key <_key-id_> md5 <_password_>
```

- The key-id should be between 1 and 65534.
- The password should be an alphanumeric string between 8 to 16 characters long.

Deleting an NTP authentication key

Delete a previously configured authentication key using the following command:

```
switch(config)# no ntp authentication-key <_key-id_>
```

Adding an NTP trusted key

Mark a previously configured authentication key as a "trusted" key using the following command:

```
switch(config)# ntp trusted key <_key-id_>
```

Deleting an NTP trusted key

Unmark a previously configured trusted key using the following command:

```
switch(config)# no ntp trusted-key <_key-id_>
```

1.17.6. Viewing NTP global information

The **show ntp status** command displays global NTP configuration information.

```
switch# show ntp status
NTP is enabled
NTP authentication is enabled
Uptime: 2 hrs
```

If the switch has synchronized its time with an NTP server, then those details are also displayed:

```
switch# show ntp status
NTP is enabled
NTP authentication is enabled
Uptime: 2 hrs
Synchronized to NTP Server 10.93.55.11 at stratum 4
Poll interval = 1024 seconds
Time accuracy is within 1.676 seconds
Reference time: Wed Jan 27 2016 19:01:48.647
```

1.17.7. Viewing NTP servers

The **show ntp associations** command displays the configured servers. The server to which the switch has synced is marked with a * in the beginning.

```
switch# show ntp associations
-----------------------------------------------------------------------
  ID                                     NAME           REMOTE  VER
-----------------------------------------------------------------------
*  1                              10.93.55.11      10.93.55.11    3
   2                     2013::42:acff:fe11:5  2013::42:acff:f    3
   3         2601:8:8a80:38:ca60:ff:fe33:c1a4  2601:8:8a80:38:    3
-----------------------------------------------------------------------
-----------------------------------------------------------------------
KEYID         REF-ID  ST  T  LAST  POLL  REACH    DELAY  OFFSET  JITTER
-----------------------------------------------------------------------
  -     16.77.112.60   4  U    21    64      1   14.754  62.274   0.001
  -           .INIT.  16  U     -    64      0    0.000   0.000   0.000
  -    252.81.37.247   5  U     3    64      3    0.211   0.062   0.031
-----------------------------------------------------------------------
```

Note that under the "NAME" column, only the first 15 characters of the server name are displayed.

1.17.8. Viewing NTP authentication keys

The **show ntp authentication-keys** command displays the configured authentication keys.

```
switch# show ntp authentication-keys
---------------------------
Auth-key       MD5 password
```

```
---------------------------
       2     aNicePassword2
       3      aNewPassword3
---------------------------
```

1.17.9. Viewing NTP trusted keys

The **show ntp trusted-keys** command displays the configured trusted authorization keys.

```
switch# show ntp trusted-keys
------------
Trusted-keys
------------
2
------------
```

1.17.10. Viewing NTP statistics

The **show ntp statistics** command displays the NTP statistics.

```
switch# show ntp statistics
            Rx-pkts    224513
    Cur Ver Rx-pkts    146
    Old Ver Rx-pkts    0
         Error pkts    0
   Auth-failed pkts    0
      Declined pkts    0
    Restricted pkts    0
  Rate-limited pkts    0
           KOD pkts    0
```

1.17.11. Troubleshooting NTP

Scenario 1

The NTP association is stalled in the .INIT. state:

```
switch# show ntp associations
-----------------------------------------------------------------------
  ID                                     NAME           REMOTE  VER
-----------------------------------------------------------------------
*  1                              10.93.55.11      10.93.55.11    3
   2                     2013::42:acff:fe11:5  2013::42:acff:f    3
   3         2601:8:8a80:38:ca60:ff:fe33:c1a4  2601:8:8a80:38:    3
-----------------------------------------------------------------------
-----------------------------------------------------------------------
KEYID         REF-ID  ST  T  LAST  POLL  REACH    DELAY  OFFSET  JITTER
-----------------------------------------------------------------------
  -     16.77.112.60   4  U    21    64      1   14.754  62.274   0.001
  -           .INIT.  16  U     -    64      0    0.000   0.000   0.000
  -    252.81.37.247   5  U     3    64      3    0.211   0.062   0.031
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

If authentication is not enabled: - reverify if the server is reachable (Rx packets are incrementing in show ntp statistics output) - reverify if the version mentioned for the server is supported by the server

If authentication is enabled: - reverify if the server is reachable (Rx packets are incrementing in show ntp statistics output) - reverify if the version mentioned for the server is supported by the server - reverify if the key mentioned is the one that has been previously configured on the server - reverify if the password mentioned while configuring the authentication-key matches too - reverify if this key has been marked as "trusted"

1.17.12. Generic tips for troubleshooting

Depending on the NTP server from which the switch is trying to synchronize time, it can take awhile before the transaction is successful. Time taken for synchronization can vary from 64 to 1024 seconds.

The show ntp statistics output should show an increase in Rx packets for some activity to proceed. An increase in packet drops suggests erroneous configurations or conditions. For example:

- Rx-pkts - Total NTP packets received.
- Cur Ver Rx-pkts - Number of NTP packets that match the current NTP version.
- Old Ver Rx-pkts - Number of NTP packets that match the previous NTP version.
- Error pkts - Packets dropped due to all other error reasons.
- Auth-failed pkts - Packets dropped due to authentication failure.
- Declined pkts - Packets denied access for any reason.
- Restricted pkts - Packets dropped due to NTP access control.
- Rate-limited pkts - Number of packets discarded due to rate limitation.
- KOD pkts - Number of Kiss of Death packets sent.

When the switch successfully synchronizes to a server, the corresponding status is displayed as part of show ntp status output.

The REF-ID field in the output for show ntp associations carries suggestive values. For the list of supported values, see http://doc.ntp.org/4.2.6p5/refclock.html.

For general NTP debugging information, see http://doc.ntp.org/4.2.6p5/debug.html. http://doc.ntp.org/4.2.8p8/monopt.html.

For information about the RFC for NTPv6 REF-ID format, click here.

1.18. Mirror feature

1.18.1. Overview

The port mirroring feature enables traffic on one or more switch interfaces to be replicated on another interface.

1.18.2. Mirror session

A mirror session defines the settings for the replication of data between one or more source interfaces and a destination interface.

A maximum of four mirror sessions can be active at the same time on the switch. There is no limit on the number of inactive sessions that can be defined in the configuration.

Each mirror session has a single output, or *destination* interface, and zero or more input, or *source* interfaces. The destination interface is the recipient of all mirrored traffic, and must able to support the combined data rate of all source interfaces. Source interfaces can be configured to mirror received traffic, transmitted traffic, or all traffic.

Source and destination interfaces do not need to reside in the same subnet, VLAN or VRF.

A LAG can be specified as either a source or destination interface. The switch internally handles the mirroring of the traffic appropriately across all the LAG member interfaces.

Mirroring is VRF agnostic. That is, a network administrator may choose to specify source interfaces from different VRFs in the same mirror session and have a single destination for the mirrored traffic.

1.18.3. Mirror rules

The following rules apply when creating a mirror session:

1. An interface cannot be both a source and destination in the same mirror session.

2. The destination interface in an **active** mirror session cannot be the source or destination in another **active** mirror session.

3. The source interface in an **active** mirror session cannot be the destination in another **active** mirror session.

4. The destination interface cannot be a member of a VLAN nor have an IP address configured.

5. The destination interface cannot have the spanning tree protocol enabled on it.

Note:

- If you try to activate a mirror session that violates rules 2 or 3 it will remain shutdown.

- The same interface can be the source in more than one mirror session as long as it does not violate rule 1 or 3.

- You can configure multiple session that violate rules 3 or 4 as long as they are not active at the same time.

1.18.4. Creating a new mirror session

- Change to configuration mode.

```
switch# configure terminal
```

- Create a new mirror session. In the following example, the session is called mirror_3.

```
switch# (config)# mirror session mirror_3
```

- Set interface 1 as the destination. Interface 1 must be a valid configured switch interface.

```
switch# (config-mirror)# destination interface 1
```

- Add source interface. In the following example, interface 2 will only mirror incoming traffic. Interface 2 must be a valid configured switch interface.

```
switch# (config-mirror)# source interface 2 rx
```

- Add source interface. In the following example, interface 3 will only mirror outgoing traffic. Interface 3 must be a valid configured switch interface.

```
switch# (config-mirror)# source interface 3 tx
```

- Add source interface. In the following example, interface 4 will mirror all traffic. Interface 4 must be a valid configured switch interface.

```
switch# (config-mirror)# source interface 4 both
```

- Activate the mirror. Prior to mirror activation ensure that all configured interfaces are also actived or results may not be as expected.

```
switch# (config-mirror)# no shutdown
```

- View mirror status to verify activation.

```
switch# (config-mirror)# do show mirror
name                                                   status
------------------------------------------------------ --------------
mirror_3                                               active
```

- View detailed mirror status to verify all interfaces.

```
switch# (config-mirror)# do show mirror mirror_3
Mirror Session: mirror_3
Status: active
Source: interface 4 both
Source: interface 2 rx
Source: interface 3 tx
Destination: interface 1
Output Packets: 143658
```

```
  Output Bytes: 1207498
```

1.18.5. Editing an existing mirror session

Mirror sessions can be modified while active.

1.18.6. Removing the destination interface

Removing the destination interface from an active mirror results in immediate shutdown. For example:

```
switch# (config)# mirror session mirror_3
switch# (config-mirror)# no destination interface
switch# (config-mirror)# do show mirror
```

```
name                                                   status
------------------------------------------------------ --------------
mirror_3                                               shutdown
```

1.18.7. Modifying the destination interface

A destination interface can be modified without removing the existing definition. The mirror session remains active and traffic is immediately sent to the new interface.

```
switch# (config)# mirror session mirror_3
switch# (config-mirror)# destination interface 5
switch# (config-mirror)# do show mirror mirror_3
Mirror Session: mirror_3
Status: active
Source: interface 4 both
Source: interface 2 rx
Source: interface 3 tx
Destination: interface 5
Output Packets: 143658
Output Bytes: 1207498
```

1.18.8. Modifying a source interface

To change the settings for a source port, re-issue the source command with the new settings. For example, if interface 3 is set to mirror both types of traffic, the following command changes it to only mirror transmitted traffic.

```
switch# (config)# mirror session mirror_3
switch# (config-mirror)# source interface 3 tx
```

1.18.9. Removing a source interface

The following example removes source interface 2 from the mirror_3 session.

```
switch# (config)# mirror session mirror_3
switch# (config-mirror)# no source interface 2
```

1.18.10. Deactivating a mirror session

```
(config)# mirror session mirror_3
(config-mirror)# shutdown
```

1.18.11. Removing a mirror session

This can be performed on both active and shutdown sessions. If the session is active, mirroring of traffic stops immediately.

```
switch(config)# no mirror session mirror_3
```

1.18.12. Displaying mirror session status

Displaying a list of all configured mirror sessions

```
swtich (config)# show mirror
```

```
name                                                            status
--------------------------------------------------------------- --------
mirror_1                                                        active
mirror_2                                                        shutdown
mirror_3                                                        active
```

1.18.13. Displaying detailed mirror session information

```
switch(config)# show mirror mirror_3
Mirror Session: xyz
Status: active
Source: interface 4 both
Source: interface 2 rx
Source: interface 3 tx
Destination: interface 1
Output Packets: 143658
Output Bytes: 1207498
```

1.18.14. Troubleshooting

Unable to add interface to mirror

When attempting to add a source or destination interface to a mirror session, if the message *Invalid interface <interface_name>* is seen, make sure the interface has been added to the switch configuration via the *interface <interface_name>* command. Note that the interface must also be activated via the *no shutdown* command for an active mirror session to function properly.

No data is being mirrored

- Verify physical connectivity of the source and destination interfaces.

- Verify that all member interfaces have been activated via the *no shutdown* command
- Display details of the mirror session to ensure the desired ports are added and the session status is active.
- Since session activation, if any source/destination LAG interface membership changes have occurred, or if any interface routing changes have occurred, these interfaces will need re-adding to the mirror. If one such interface was the session destination, the session will also need to be reactivated.

Too much or not enough data is seen on the destination interface

Ensure that the source port you added is configured for the correct direction: receive, transmit, or both.

Many to many condition

It is best that a source interface is in only one active mirror session. It is permissible for the same source interface to be in difference sessions. When the same source interface(s) are in multiple mirror sessions, there is additional restriction that any one mirror session cannot have sources that are also sources in different sessions. This is called a ‘many-to-many’ situation.

An example of a many-to-many situation: Three mirror sessions A, B, & C. Interface 1 is a source in session A Interface 2 is a source in sessions A and B Interface 3 is a source in sessions B and C Interface 4 is a source in session C

Mirror session B will not work because its sources (2 & 3) are in a many-to-many relationship with other mirror sessions.

1.19. CLI support for Autoprovision

1.19.1. Overview

Autoprovisioning (a.k.a Zero Touch Provisioning - ZTP) is a feature that enables automatic provisioning of switch when it is deployed. Using a DHCP option advertised by DHCP server in the setup, the switch downloads a provisioning script and executes it. The provisioning script can do many things, such as new management users, downloading ssh keys, installing a server certificate, etc. This feature is mainly used to download ssh keys and add user ids to the switch enabling key based authentication of management users.

1.19.2. Setting up the basic configuration

The feature is enabled by default and cannot be turned off through CLI. To disable the autoprovisioning feature, configure the DHCP server to NOT send option 239 in the DHCP reply/ack messages.

1.19.3. Verifying the configuration

Not applicable.

1.19.4. Writing autoprovisioning script

- Shell, Perl and Python scripts are supported as auto-provisioning script.
- It is assumed that the first line of the script contains entries like #!/bin/sh to distinguish between a shell, perl and python scripts.
- The provisioning script must contain a line OPS-PROVISIONING as a comment. This is for a very rudimentary validation that we got a valid provisioning script.
- The script is executed on the Linux shell(Bourne shell) of the switch.

1.19.5. Major Executables

In the script the following executables can be used to configure CLI commands in the switch(VTYSH), to download other files from outside the switch(WGET) and to log the appropriate ZTP messages(LOGGER).

```
VTYSH=/usr/bin/vtysh
WGET=/usr/bin/wget
LOGGER=/usr/bin/logger
```

Other usual shell commands which are available in OpenSwitch can also be used in an autoprovisioning script.

Examples

Suppose we want to configure the below CLI commands and copy some SSH Public keys from outside on to the switch:

```
vlan 2
   no shutdown
interface 3
   no shutdown
   no routing
   vlan trunk native 1
   vlan trunk allowed 2
```

VTYSH

- CLI commands can be configured with the executable /usr/bin/vtysh.
- The arguments to this executable are the exact CLI commands in a correct order with a tag of -c preceding each command.
- To configure the above CLI commands in switch, execute the below commands in switch's Linux shell:

```
root@switch:~# VTYSH=/usr/bin/vtysh
root@switch:~# $VTYSH -c "configure terminal" -c "vlan 2" -c "no shutdown"
-c "interface 3" -c "no shutdown" -c "no routing" -c "vlan trunk allowed 2"
-c "exit"
```

- The above commands can also be incorportated in an autoprovisioning shell script the example of which is given below.
- Since autoprovisioning script will get executed on the Linux shell so automating a script to read the CLI configuration from a separately downloaded config file and run the CLI commands like above is fairly easy task.

WGET

- Use /usr/bin/wget to download config file/SSH keys.
- Shell commands like wget can be used to download any type of file into switch.

```
$WGET <Link to config/SSH key file> -O <File-name>
```

LOGGER

- Logging events with /usr/bin/logger:

```
$LOGGER -i "MESSAGE that needs to be documented"
```

Sample script file

```
#!/bin/sh
```

```
# OPS-PROVISIONING
# Sample auto-provisioning script for OpenSwitch.
```

```
# executable files used by this script
VTYSH=/usr/bin/vtysh
WGET=/usr/bin/wget
LOGGER=/usr/bin/logger
```

```
# Run vtysh command on switch Linux shell
$VTYSH -c "configure terminal" -c "vlan 2" -c "no shutdown"  -c "interface 3"
-c "no shutdown" -c "no routing" -c "vlan trunk allowed 2" -c "exit"
```

```
# Log the events
$LOGGER -i "Configured vlan 2"
$LOGGER -i "Configured interface 3 with vlan trunk 2"
```

```
# Create a new user or copy SSH keys to an existing user home directory
# Download SSH keys from outside the switch and save it for username "netop"
$WGET http://192.168.1.1/sshkeys.txt -O /home/netop/.ssh/authorized_keys
[ $? -eq 0 ] && $LOGGER -i "Copied and saved authorized SSH keys for netop"
```

```
<---- Complete the script to read configuration file and apply any other
commands on switch using VTYSH ---->
```

1.19.6. Troubleshooting the configuration

Condition

Autoprovisioning is not performed.

Cause

- The DHCP option 239 not configured on the DHCP server.

Verify whether DHCP server is correctly configured.

- The URL in the DHCP option 239 is not valid.

The URL can be seen in the output of "show autoprovisioning" command. If this URL is incorrect, autoprovisioning does not happen.

- Provisioning script does not contain the line OPS-PROVISIONING.

Please verify the provisioning script.

1.20. System function

1.20.1. Overview

This guide provides the detail for managing and monitoring platform components on the switch. All the configuration commands work in configure context. Some of the following switch physical components are:

- Fans
- Temperature sensors
- Power supply modules
- LED This feature allows you to configure fan speed or LED state.

1.20.2. Configuring system

Setting up the basic configuration

Setting the fan speed

Fans can be configured to operate at a specified speed with the *fan-speed (slow medium* | *fast* | *max)* commands. By default all the fans operate at normal speed and change according to the system temperature.

```
switch# configure terminal
switch(config)# fan-speed slow
switch(config)#
```

Setting LED state

The *led led-name (on* | *off* | *flashing)* command lets the user to set the state of the LED . By default all the LEDs will be in off state. User should know the name of the LED of whose state is to be set. In the example below *base-loc* LED is set to on.

```
switch# configure terminal
switch(config)# led base-loc on
switch(config)#
```

Setting the time zone

The *timezone set TIMEZONE* command lets the user set the time zone on the switch. By default the time zone on the switch is UTC (Coordinated Universal Time). All the time zones available in the posix standard are supported through this command. In the example below the *us/alaska* time zone is set on the system.

```
switch(config)# timezone set u?
uct                us/arizona         us/hawaii          us/pacific
universal          us/central         us/indiana-starke  us/samoa
us/alaska          us/east-indiana    us/michigan        utc
us/aleutian        us/eastern         us/mountain
```

```
switch(config)# timezone set us/alaska
```

1.20.3. Verifying the configuration

View the fan information.

show system fan command displays the detailed information of fans in the system.

```
switch#show system fan
Fan information
...............................................................
Name     Speed  Direction      Status        RPM
...............................................................
base-2R  slow  front-to-back  ok            5700
base-5L  slow  front-to-back  ok            6600
base-3L  slow  front-to-back  ok            6600
base-4L  slow  front-to-back  ok            6600
base-5R  slow  front-to-back  ok            5700
base-2L  slow  front-to-back  ok            6650
base-3R  slow  front-to-back  ok            5700
base-1R  slow  front-to-back  ok            5750
base-1L  slow  front-to-back  ok            6700
base-4R  slow  front-to-back  ok            5700
..............................................................
Fan speed override is set to : slow
..............................................................
```

View the LED information

The *show system led* command displays LED information.

```
switch#sh system led
Name           State     Status
.....................................
base-loc       on       ok
```

View power-supply information

The *show system power-supply* command displays detailed power supply information.

```
switch#sh system power-supply
Name           Status
............................
base-1         ok
base-2         Input Fault
```

View temperature sensor information

The *show system temperature* command displays temperature sensor information.

```
switch#sh system temperature
Temperature information
....................................................
```

```
            Current
Name     temperature    Status         Fan state
            (in C)
....................................................
base-1    22.00          normal         normal
base-3    18.50          normal         normal
base-2    20.50          normal         normal
```

View detailed version information

The *show version detail* command displays the version, source type, and source URL of every package present in the switch image.

```
switch#show version detail
PACKAGE     : kernel-module-gspca-spca1528
VERSION     : 3.14.36+gitAUTOINC+a996d95104_dbe5b52e93
SOURCE TYPE : git
SOURCE URL  : https://git.yoctoproject.org/linux-yocto-3.14.git;bareclone=1;branch=s
```

```
PACKAGE     : python-jsonpatch
VERSION     : 1.11
SOURCE TYPE : http
SOURCE URL  : http://pypi.python.org/packages/source/j/jsonpatch/jsonpatch-1.11.tar.
```

```
PACKAGE     : ops-cli
VERSION     : a70df32190755dabf3fb404c3cde04a03aa6be40~DIRTY
SOURCE TYPE : other
SOURCE URL  : NA
```

```
PACKAGE     : dbus-1
VERSION     : NA
SOURCE TYPE : other
SOURCE URL  : NA
```

View time zone information

The *show system timezone* command displays the timezone information on the switch. By default the timezone configured should be UTC.

```
switch# show system timezone
System is configured for timezone : UTC
     DST active: n/a
```

```
switch# show date
Thu Jun 16 00:08:26 UTC 2016
```

If the time zone is configured for "US/Alaska", then it may be verified with the following commands:

```
switch# show running-config
Current configuration:
!
Version 0.1.8
timezone set us/alaska
!
```

```
!
!
!
vlan 1
   no shutdown
```

```
switch# show system timezone
System is configured for timezone : US/Alaska
     DST active: yes
Last DST change: DST began at
                 Sun 2016-03-13 01:59:59 AKST
                 Sun 2016-03-13 03:00:00 AKDT
Next DST change: DST ends (the clock jumps one hour backwards) at
                 Sun 2016-11-06 01:59:59 AKDT
                 Sun 2016-11-06 01:00:00 AKST
```

1.21. Control Plane Policing

1.21.1. Overview

Control plane policing (CoPP) in OpenSwitch is used to prioritize traffic handling by the CPU, and to protect the switch from DoS attacks.

1.21.2. Packet classes and rate limiting on Broadcom-based platforms

The following default settings are defined during switch initialization and cannot be changed.

Priority	CPU Queue	Description
Critical	Q10	xSTP
Important	Q9	OSPF,BGP
LLDP/LACP	Q8	LLDP, LACP
Management	Q7	Currently, inband management traffic is not supported.
Unknown IP	Q6	Unknown destination IP/IPv6
SW-PATH	Q5	Unicast ARP, Unicast ICMP, ICMP, ICMPv6, IP options
Normal	Q4	Broadcast ARP, ICMP, DHCP, Broadcast/Multicast
sFlow	Q3	Sampled sFlow traffic
Snooping	Q2	
Default	Q1	Unclassified packets
ACL Logging	Q0	ACL logging

Packet Class	Description	Queue	Rate Limit (PPS)
ACL_LOGGING	ACL Logging	Q0	5
ARP_BC	Broadcast ARP Packets	Q4	1000
ARP_UC	Unicast ARPs	Q5	1000
BGP	BGP packets	Q9	5000
DHCP	DHCP packets	Q4	500
DHCPV6	IPv6 DHCP packets	Q4	500
ICMP_BC	IPv4 broadcast/multicast ICMP packets	Q4	1000
ICMP_UC	IPv4 unicast ICMP packets	Q5	1000
ICMPV6_MC	IPv6 multicast ICMP packets	Q4	1000
ICMPV6_UC	IPv6 unicast ICMP	Q5	1000
IPOPTIONV4	Packets with IPv4 options	Q5	250
IPOPTIONV6	Packets with IPv6 options	Q5	250

Packet Class	Description	Queue	Rate Limit (PPS)
LACP	LACP packets	Q8	1000
LLDP	LLDP packets	Q8	500
OSPF_MC	Multicast OSPF packets	Q9	5000
OSPF_UC	Unicast OSPF packets	Q9	5000
sFlow	Sampled sFlow packets	Q3	5000
STP	STP packets	Q10	1000
UNKNOWN_IP_DEST	Unknown IPv4 or Ipv6 destination/Glean packets	Q6	2500
UNCLASSIFIED	Unclassified packets	Q1	5000

1.21.3. Monitoring control plane policing

Use the following commands to monitor CoPP packet statistics/status for each packet class:

```
switch# show copp statistics icmpv4-unicast
       Control Plane Packet: ICMPv4 UNICAST packets

rate (pps):                  1000
burst size (pkts):           1000
local priority:                 5

packets passed:                 7    bytes passed:               786
packets dropped:                0    bytes dropped:                0

switch# show copp statistics
       Control Plane Packets Total Statistics

total_packets_passed:        6967    total_bytes_passed:       650274
total_packets_dropped:          0    total_bytes_dropped:           0

Control Plane Packet: BGP packets

rate (pps):                  5000
burst size (pkts):           5000
local priority:                 9

packets passed:                 0    bytes passed:                 0
packets dropped:                0    bytes dropped:                0

Control Plane Packet: LLDP packets

rate (pps):                   500
burst size (pkts):            500
local priority:                 8

packets passed:                 0    bytes passed:                 0
packets dropped:                0    bytes dropped:                0

...
```

If a packet class is not supported on the switch, then all status and statistics values for that class will be empty. Statistics that are not supported for a packet class will be empty.

1.22. CLI support for Config Persistence

1.22.1. Overview

There are two types of configurations: the current running configuration and the startup configuration. The running configuration is not persistent across reboots but the startup configuration is persistent across reboots. While there are currently no provisions to facilitate rollbacks or local preservation of old configurations, these functions are being investigated as further enhancements.

1.22.2. Configuration types

- Running: The running configuration is dynamic and is the current state of all config type elements in the OVSDB.
- Startup: If present, the startup configuration will be used on the next boot.

1.22.3. Startup configuration persistence

When a configuration is saved as a startup configuration, it is stored in the configtbl table in OVSDB. The confgtbl can have from zero to multiple rows.

The content of each row includes:

- type: Only supported type is startup.
- name: Unique name, specified either by the system or by the user (Currently not populated).
- writer: Identifies who requested this configuration to be saved (Currently not populated).
- date: Date/Time when this row was last modified (Currently not populated).
- config: Configuration data.
- hardware: JSON formatted list of dictionaries containing the following information for all subsystems configured by the configuration data (Currently not populated).

1.22.4. Configuration format

The configuration data is stored as a JSON string. The schema used is the same schema used by the REST API.

1.22.5. Action taken during configuration save

When the user requests that the running configuration be saved as a startup-config, the following actions are taken:

- All config type OVSDB elements are extracted from the database and formatted into the configuration format as noted below.
- If a startup row is not found in the configdb, create a new row with type=startup or overwrite the existing row and save the configuration to config row.

- Configtbl is updated with all required information.

1.22.6. System action taken during system boot

Except for the configurations table, the OVSDB is not persisted across reboots, so that it is initially empty. After the platform daemons discover all present hardware and populate the OVSDV with relevant hardware information, the configuration daemon (cfgd) checks to see if any saved configuration exists by checking for a startup type entry. If a startup configuration is found, it is applied to the remaining tables. If a startup configuration is not found, cfgd notes the startup configuration was not found.

1.22.7. User actions

The REST and CLI APIs provide rich commands for managing the configuration of the system, allowing the user to:

- Request that the current running configuration be saved as a startup configuration.
- Request that a startup configuration be written to the running configuration.
- Request a read/show of a startup configuration.
- Request a read/show of a running configuration.

1.23. Logrotate

1.23.1. Overview

Logrotate rotates and compresses the log files either based on period or based on size or based on both period and size (whichever condition triggers first). Rotated log files are stored locally or transferred to the remote destination.

1.23.2. How to use the feature

With no initial configuration , logrotate runs as an hourly cron job from the /etc/cron.hourly path with the following default configuration,

```
h1# show logrotate
Logrotate configurations :
Period            : daily
Maxsize           : 10MB
```

All rotated logs are stored locally.

As per this configuration, the default behavior is that the log rotation feature rotates the log files daily. Log rotation is also triggered if the maximum file size exceeds 10 MB. Out of the period and size, whichever condition occurs first, triggers the log rotation.

1.23.3. Changing the default threshold values for log rotation

Use the **logrotate** CLI command to change the default threshold values for period and file size.

```
logrotate period (hourly| daily| weekly | monthly )
logrotate maxsize <1-200>
```

1.23.4. Setting up the optional configuration

Identify a remote host for receiving rotated log file by using the following CLI command:

```
logrotate target  { tftp://A.B.C.D | tftp://X:X::X:X }
```

Only the TFTP protocol is supported for remote transfer. Both IPv4 and IPv6 host addresses are supported.

1.23.5. Verifying the configuration

- Verify log rotation parameters by running the following command:

  ```
  h1# show logrotate
  Logrotate configurations :
  Period            : weekly
  Maxsize           : 20MB
  ```

```
Target              : tftp://2001:db8:0:1::128
```

- Verify Log-rotate configuration CLIs with the show running-config command.

1.23.6. Troubleshooting the configuration

Condition 1

Log-rotation is not happening regardless of *period* value.

Cause

Log-rotate does not happen for empty files (i.e if file size is zero)

Remedy

Log-rotate happens when file size is greater than zero

Condition 2

Rotated log files are not transferred to remote host

Cause

1. The remote host might not be reachable

2. The Tftp server on the remote host might not have sufficient privileges for file creation

Remedy

1. Verify that the remote host is reachable.

2. Ensure that the TFTP server is configured with the required file creation permissions.

   a. For example, in tftpd-hpa server, change the configuration file in /etc/default/tftpd-hpa to include -c in TFTP_OPTIONS. (for example, TFTP_OPTIONS="--secure -c").

Condition 3

Log rotation does not occur immediately after the maximum file size for the log file is reached.

Cause

The log rotation checks the size of the file on the first minute of every hour. If the maximum file size is reached in the meantime, the log rotation does not occur until the next hourly check of the file size.

Remedy

Log rotation is working as designed. The log rotation feature is designed to check the file size on an hourly basis.

1.24. Show Tech

1.24.1. Overview

The Show Tech Infrastructure is used to collect a summary of switch or feature specific information. This Infrastructure runs a collection of show commands and produces the output in text format. The collection of show commands per feature are present in the show tech configuration file, accordingly show tech displays those commands for the given feature. The output of the show tech command is mainly useful to analyze system or feature behavior. Tools can be developed to parse show tech command output in order to arrive at a meaningful conclusion and aid in troubleshooting.

1.24.2. How to use the Show Tech feature

- To collect the switch wide show tech information, run the cli command show tech.
- To collect a feature specific show tech information, run the cli command show tech FEATURE-NAME.
- To display the list of features supported by show tech, run the cli command show tech lis.

1.24.3. Setting up the basic configuration

The Show Tech infrastructure loads its configuration from the show tech configuration yaml file located in (/etc/openswitch/supportability/ops_showtech.yaml). This file contains the default configuration for show tech.

1.24.4. Verifying the configuration

Execute the cli command **show tech list** and verify the features are listed as configured.

1.24.5. Troubleshooting the configuration

Condition

The show tech cli commands result in the following error:

```
Failed to obtain Show Tech Configuration
```

Cause

This "Failed to obtain Show Tech Configuration" error can appear in the following two cases:

- The ops_showtech.yaml configuration file is missing in its path
- The ops_showtech.yaml configuration file is wrongly configured.

Remedy

1. Ensure that the ops_showtech.yaml file is present in its path (/etc/openswitch/supportability/ops_showtech.yaml).

2. Verify that the ops_showtech.yaml configuration file is a valid yaml file, using the yaml lint tools.
3. Verify that the structure of the configuration is valid. Refer here for structure information and examples.

1.25. Show Vlog

1.25.1. Overview

The show vlog command is used to display vlog messages of ops daemons.

The show vlog config command is used to display a list of features and corresponding daemons log levels of syslog and file destinations. The show vlog config command is useful to generate switch feature vlog configuration log levels for administrators, developers, support, and lab staff.

1.25.2. Show vlog command to display vlogs

The show vlog command is used to display vlog messages of ops daemons. There are various ways to filter the vlog messages.

1.25.2.1. Filter vlogs based on daemon name

The show vlog daemon <daemon_name> command is used to filter the vlog messages based on a daemon-name/word. This command displays only vlog messages for the specified daemon-name/word.

1.25.2.2. Filter vlogs based on severity

The show vlog severity <emer/err/warn/info/debug> command is used to filter and display the vlog messages of a specified severity level and above.

1.25.3. Using the show vlog command to access feature/daemon information

There are various ways to access supported feature information using the show vlog config command.

1.25.3.1. List vlog supported features

Use the show vlog config list command to get the list of supported features on the console with descriptions.

1.25.3.2. Severity log level of a feature

Use the show vlog config feature <feature_name> command to capture log levels of file and syslog destinations on the console for a specific feature. Severity log levels of supported features

Use the show vlog config command to get the list of supported features, and corresponding daemons' log levels of file and syslog destinations, on the console.

1.25.3.3. Severity log level of a daemon

Use the show vlog config daemon <daemon_name> command to capture log levels of file and syslog destinations on the console for the specified daemon.

1.25.3.4. Configure feature logging level and destination

Configure the switch using the vlog (feature|daemon) <name> <syslog/file/all> <emer/err/warn/info/dbg> command in configuration mode. Use the show vlog config feature <feature_name> and show vlog config daemon <daemon_name>command to obtain the corresponding feature/daemon log level changes of file and syslog destinations on the console.

1.25.4. Troubleshooting the configuration

Condition

The show vlog config command results in the following error *Failed to capture vlog information <feature/daemon>*

Configuration file is missing

If the *error ops_featuremapping.yaml configuration file is missing in its path* message appears, ensure that the *ops_featuremapping.yaml* file is present in the */etc/openswitch/supportability/ops_featuremapping.yaml* path.

Configuration file is not properly configured

If the error *ops_featuremapping.yaml configuration file is wrongly configured* message appears, use the yaml tools to confirm that the configuration file (ops_featuremapping.yaml) is valid.

1.25.5. Feature to daemon mapping

The */etc/openswitch/supportability/ops_featuremapping.yaml* file contains feature to daemon mapping configurations.

1.26. BroadView

Broadview support depends on the OpenNSL library.

1.26.1. Overview

The Broadcom BroadView software suite provides visibility into Broadcom switch silicon, exposing various instrumentation capabilities.

The software suite consists of an agent that runs on the switch, and an application that runs on a remote device. The application communicates with the agent using the Open REST API. The application analyzes and visualizes data provided by the agent, and enables administrators to fine-tune BST parameters.

In OpenSwitch, the agent is implemented by the BroadView daemon. The daemon provides instrumentation capability for OpenSwitch. In the current release, it obtains MMU buffer statistics from Broadcom silicon and exports them via the REST API. This allows applications (instrumentation collectors) to obtain MMU Buffer statistics, visualize buffer utilization patterns, and detect microbursts. This information provides administrators with visibility i nto the network and switch performance, and lets them fine-tune the network.

The BroadView daemon is recommended for use with OpenSwitch running on a hardware platform. The daemon runs as a background process in OpenSwitch.

1.26.2. Prerequisites

An application must be installed on a remote device to retrieve MMU buffer statistics from the agent on the switch via the REST API.

1.26.3. Basic configuration

1. Configure the IP address of the remote device running the application which will retrieve the statistics with the command:

   ```
   broadview client ip <ip-address> port <port-num>
   ```

   For example:

   ```
   switch(config)# broadview client ip 10.130.168.30 port 8080
   ```

2. Configure the port on which the agent will communicate with the remote device with the command:

   ```
   broadview agent-port <port-num>
   ```

   For example:

   ```
   switch(config)# broadview agent-port 8080
   ```

3. Verify the configuration with the command:

   ```
   show broadview
   ```

For example:

```
switch# show broadview
BroadView client IP is 10.130.168.30
BroadView client port is 8080
BroadView agent port is 8080
```

1.27. Incremental software upgrade

From the release 2.0.0, a .deb package is provided for users; the user can use it to upgrade the NOS without rebooting the whole system. ONIE tools are not in use as well.

Example:

```
admin@switch:~# sudo -i
root@switch:~# dpkg -i ops-package_2.0.1_amd64.deb
```

1.28. Text-based configuration

Users can create text-based configurations outside of a switch, and load it back to the device.

The configuration, startup-config, is located in */var/local/openvswitch*

1.29. OpenSwitch Web User Interface (UI)

1.29.1. Overview

The OpenSwitch web UI provides an easy-to-see visual representation displaying the state of the switch. Easy-to-use view and configuration screens help the user to understand and configure complex features.

1.29.2. Accessing the web UI

To access the web UI, open a web browser (Google Chrome preferred) and enter the IP address of the switch management interface.

The web server (REST daemon) is disabled by default. To activate it, please do the following:

```
admin@switch:~# sudo -i
root@switch:~# systemctl enable restd
root@switch:~# systemctl start restd
```

1.29.3. Login

Web UI IP address is the same as the management port address.

When accessing the switch web UI, the user first sees the login screen. Default credentials are "netop/netop."

1.29.4. Overview

The Overview screen displays important information and statistics about the switch.

System

The system panel includes information about the switch, such as:

- the product name
- serial number
- vendor
- version
- ONIE Version
- base MAC address

General

The features panel includes:

- a listing of the switch features and their current state (enabled or disabled)

- number of VLANs configured on the switch
- number of interfaces on the switch
- the maximum transmission unit (MTU)
- the maximum interface speed

Hardware

The hardware panel shows the status of:

- the power supplies
- temperatures
- fans

Top Interface Utilization panel

The Top Interface Utilization panel displays the ports with the top utilization (either through transmitting or receiving data). The list automatically sorts, moving the port with the top utilization percentage to the top. Clicking the graph icon at the top of the gauge takes you to the Interfaces Monitor screen, which displays details for the top interface.

Log

The log panel displays the last system log messages.

1.29.5. Interfaces

The Interfaces screen displays the box graphic of the switch, interface table with details, edit and search options. The Details option is selected by default. The total number of rows in the table is displayed in the top-left corner of the table. The total dynamically updates by showing the number of results found by the search.

If you select an interface in the table and the Details option is selected, additional details about the selected interface are displayed in the Interface Details panel on the right side of the screen.

1.29.6. Interface Details panel

The Interface Details panel provides configuration and health information for an interface, in addition to providing troubleshooting information via the LLDP “map” of directly connected devices.

The Interface Details panel has three tabs:

- General: The General tab has information about the interface configuration.
- Statistics: All port statistics are present in the Statistics tab.
- LLDP: LLDP neighbor information and statistics are provided under the LLDP tab.

If an interface can be split, the split children and split parent details are displayed in the Interface Details panel.

To close the Interface Details panel, click the close (X) icon in the top-right corner.

Edit icon (wrench)

The edit icon (wrench) on the Interfaces screen is enabled when you select a row on the table or click a port on the box graphic.

Edit panel

When you click the edit icon, the Edit panel appears with the current:

- admin state
- auto negotiation
- duplex and flow control of the interface, which can be configured

Click the “OK” button in the bottom of the Edit panel. Click the Close (X) icon at the top-right corner of the Edit panel to close the panel.

Split/Unsplit option

When editing a split parent interface, the Split/Unsplit option becomes available. If the split parent interface is configured split from here, the table updates to show all the split interfaces of that parent. The split parent cannot be configured until it is unsplit.

1.29.7. LAGs

The Link Aggregation (LAG) screen displays the LAG table with details, edit, add, delete and search options. The Details panel is displayed by first selecting a LAG in the list number and then selecting the Details option. The number of LAGs are displayed in the top-left corner of the table. The total updates dynamically by showing the number of results found by the search.

1.29.7.1. Adding LAGs

Click the plus sign (+) to add a LAG. The software displays the LAG add panel with 2 tabs:

- ID & Interfaces
- Attributes

ID & Interfaces

The ID & Interfaces tab has an input box for the LAG ID to be created. The list of available LAG ID ranges is shown. There are two icons plus (+) and minus (-) which increment and decrement the LAG ID. The software provides two list for interfaces: available and those currently part of that LAG. The boxes next to "Available" and the LAG name select and deselect everything in the list.

Adding an interface

To add an interface:

- Select the interface ID from the Available list.

- Click the greater than sign (>).

Removing an interface

An interface can be removed from the LAG by selecting it and clicking the less than sign (<).

Attributes

LAG Attributes includes the following configuration options:

- Aggregation Mode: Aggregation Mode can be active, passive, or off (static LAG). The default setting is off.
- Rate: Rate at which LACP control packets are sent to an LACP-supported interface to detect status and fault in the LAG: — Fast LACP: LACP packets are sent every second. — Slow LACP (default): LACP packets are sent every 30 seconds.
- Fallback: Fallback is true or false (default).
- Hash: Hashing is used to determine how the LAG chooses which interface to forward traffic:
- L2 Src/Dst (source/destination): Layer 2 hashing mode.
- L3 Src/Dst (default): Layer 3 hashing mode.
- L4 Src/Dst: Layer 4 hashing mode.

1.29.7.2. Remove LAG

The currently selected LAG can be removed by clicking the minus sign (-).

1.29.7.3. Edit LAG

To change the information of an existing LAG, select the LAG and click the edit icon (wrench). The same dialog as Add LAG is displayed.

1.29.8. ECMP

The ECMP screen shows ECMP (Equal Cost Multi-Path) status and various load balancing configurations:

- Status: Determines whether ECMP is enabled in the system. Default is true
- Source IP: Determines whether the source IP participates in the ECMP hash calculation. Default is true.
- Source Port: Determines whether the source TCP/UDP port participates in the ECMP hash calculation. Default is true.
- Destination IP: Determines whether the destination IP participates in the ECMP hash calculation. Default is true.
- Destination Port: Determines whether the destination TCP/UDP port participates in the ECMP hash calculation. Default is true.

- Resilience Hashing: Determines whether the ECMP hashing preserves traffic flows when the ECMP group membership changes. Default is true.

Editing ECMP

To edit the ECMP configuration, click the wrench icon.

1.29.9. Log

The Log screen displays a table of the switch logs. By default the only the critical logs from the last hour are displayed. The fields displayed are:

- Time
- Severity
- Identifier
- Category
- Message

The Log screen provides drop-down lists for severity (Critical Only, Critical & Warning, and All) and for time (Last Hour, Last 24 Hours, and Last 7 Days). When selecting an option in the drop-down list, the software requests the switch to pull the latest logs matching the selected criteria.

In the upper-right corner of the Log screen is a search box. The search feature filters the list of logs to those matching the search criteria. All fields are used when searching.

If the length of the message field is too big to fit, the entire message is displayed when you mouse over the message field.

1.29.10. Quick Guides

Quick Guides provide online help for some of the web UI features.

1.29.11. Links

REST API

The Swagger UI link opens a new window/tab in the browser and displays the Swagger UI: https://management_interface_ip_address-or-switch_name/api/index.html

1.29.12. User

The user name of the account currently logged in is displayed at the bottom-left corner of the screen. When you click the user name, a pop-up menu with two options is displayed:

- Logout
- Change Password

Logout

It logs out the current user, and returns the user to the login page.

Change Password

The Change Password feature prompts the user for the old password, the new password and the confirmed new password.

1.30. REST API

1.30.1. Overview

The REST_API provides a management interface to interact with a switch. You can utilize the API to retrieve status and statistics information of the switch, as well as to set and change the configuration of the switch.

This feature provides two major functionalities:

- REST API service engine — Processes REST API operation requests.
- REST API documentation rendering engine — Presents a web interface documenting the supported REST API. You can interact with the REST API service engine running on the same switch through this web interface.

1.30.2. Setting up the basic configuration

This feature is disabled in the switch image build by default. This feature can be turned on as shown below:

```
systemctl enable restd
systemctl start restd
```

You do not need to do anything other than start the restd and basic network connectivity to the switch to use this feature.

1.30.3. Troubleshooting the configuration

Condition

Error in accessing the URIs supported by the REST API.

Cause

- Switch network connectivity issue
- REST daemon fails to start or has crashed

Remedy

- Ping the switch at the given IP address after making sure that the IP address is configured for the management interface of the switch.
- Make sure that the REST daemon is running.

1.30.4. Entry point

The URL for accessing REST API documentation rendered on the switch is:

```
https://management_interface_ip_address-or-switch_name/api/index.html
```

The default HTTPS port is 443. When http is used (port 80), requests get redirected to https (port 443).

To access details about the supported REST API without running a switch image, see the following website for information:

```
http://api.openswitch.net/rest/dist/index.html
```

1.30.5. CLI

This feature is an alternative to the CLI mechanism as a management interface. It has no CLIs of its own.

1.30.6. Related features

The configuration daemon and API modules utilize configuration read and write capabilities provided by this feature in the form of Python libraries.

1.30.7. Notifications

Resources can be subscribed to for monitoring resource additions, changes, and deletions that trigger notifications to the client. The client can subscribe to specific resources or a collection of resources. For example, the client can subscribe to changes for a BGP router with ASN 6001, or to a collection of BGP routers that belong to a VRF with the name vrf_default.

When a client subscribes to a specific resource, the client will be notified of its initial values, its updated values when the resource is modified, or notified when the resource is deleted. When a client subscribes to a collection of resources, the client will be notified of its initial values, and when other resources within the same collection subscription is added or deleted. Clients can have multiple subscriptions.

1.30.7.1. Subscribing

The subscribing mechanism is through REST APIs. To create a new subscription, the client must send a POST request to the URI of the subscriber. For example, if the subscriber name is "subscriber_1", the URI to send a POST request to is: https://ip_address/rest/v1/system/notification_subscriber/subscriber_1/notification_subscription

The data for the POST request contains the name of the subscription that is unique to the subscriber and a resource URI that may be for a specific resource or for a collection.

For a specific resource, the resource may look like the following: /rest/v1/system/vrfs/vrf_default/bgp_routers/6001.

For a collection, the resource may look like the following: /rest/v1/system/vrfs/vrf_default/bgp_routers

1.30.7.2. Notification message

When a monitored resource change is detected, a notification is sent to the client in a JSON formatted message, which includes the resources added, modified, or deleted. Each notification includes the values, subscription URI, and resource URI, except for a deleted resource that excludes the resource’s values.

The notification message may look like the following:

```
{
   "notifications": {
       "added": [{
           "subscription": "/rest/v1/system/notification_subscribers/
                       3562910982/notification_subscriptions/subscription_1",
           "resource": "/rest/v1/system/vrfs/vrf_default/bgp_routers/1/
                       bgp_neighbors/2.2.2.2",
           "values": {
               "remote_as": 2
           }
       }],
       "modified": [{
           "subscription": "/rest/v1/system/notification_subscribers/
                       3562910982/notification_subscriptions/subscription_2",
           "resource": "/rest/v1/system/vrfs/vrf_default/bgp_routers/1",
           "new_values": {
               "router_id": "1.1.1.1",
               "maximum_paths": 5,
           }
       }],
       "deleted": [{
           "subscription": "/rest/v1/system/notification_subscribers/
                       3562910982/notification_subscriptions/subscription_3",
           "resource": "/rest/v1/system/vrfs/vrf_default/bgp_routers/1/
                       bgp_neighbors/3.3.3.3"
       }]
   }
}
```

1.30.7.3. Notifications over WebSockets

Notifications are currently only supported over WebSockets. When a client connects to the server through WebSockets at the wss://ip_address/rest/v1/ws/notifications URI, a notification subscriber resource with a random generated subscriber name is automatically created. The subscriber name can be used for creating subscriptions through REST. When changes are detected, notifications are sent to the client through the WebSocket.

Chapter 2. Layer 2 features

2.1. Interface configuration

2.1.1. Overview

This guide provides detail for managing and monitoring the physical interface present in the switch. All configurations work in interface context. When the interface is running, all the default configurations take effect. To change the default configuration, see Setting up the basic configuration.

2.1.2. Setting up the basic configuration

1. Change to the interface context. The interface interface command changes the vtysh context to interface. The interface variable in the command depicts the name of the interface, such as interface "1" in the following example.

```
switch# configure terminal
switch(config)# interface 1
switch(config-if)#
```

2. Enable the interface. The no shutdown command enables a particular interface on the switch. Once the interface is enabled, all other configurations take effect.

```
switch(config-if)# no shutdown
switch(config-if)#
```

3. Set the interface speed. The speed command sets the interface speed. Supported speeds are 1Gbps, 10 Gbps and 40 Gbps. Depending upon the interface type, these configurations may or may not take effect.

```
switch(config-if)# speed 1000
switch(config-if)#
```

The no speed command reverts the interface speed to auto mode.

```
switch(config-if)# no speed
switch(config-if)#
```

4. Set the interface duplexity. The duplex command sets the interface duplexity to either half or full duplex.

```
switch(config-if)# duplex half
switch(config-if)#
```

The no duplex commands reverts the interface duplexity to default to full.

```
switch(config-if)# no duplex
switch(config-if)#
```

5. Set the interface MTU. The MTU of a communications protocol refers to the size in bytes of the largest frame (Ethernet) or packet (IP) that can be sent on the network. Different protocols support a variety of MTU sizes. Most IP over Ethernet implementations uses the Ethernet V2 frame

format, which specifies an MTU of 1500 bytes. Jumbo frames are Ethernet frames containing more than 1500 bytes.

```
switch(config)# mtu 2000
switch(config)#
```

Maximum configurable MTU value for jumbo frame is 9192 which allows inbound jumbo packets up to 9216(9192+padding bytes(6+6+4+4+2+2 i.e DA + SA + STAG + CTAG + LEN + FCS) bytes.

Restrictions: Egress MTU setting allows 4 additional bytes to accommodate VLAN tag. For HigGig ports(10G/40G), actual MTU value set in hardware is 4 more than what is passed from application (API).

The no mtu commands reverts the mtu of the interface to default auto mode. The *mtu auto* command sets MTU to system default.

```
switch(config-if)# no mtu
switch(config-if)#
```

6. Select the interface autonegotiation state. The autonegotiation command turns the autonegotiation state on or off. The no autonegotiation command sets the autonegotiation state to default.

```
switch(config-if)# autonegotiation on
switch(config-if)#
switch(config-if)# no autonegotiation
switch(config-if)#
```

7. Set the flowcontrol. The flowcontrol command enables the flow control mechanism (pause frame technique). The ‘no flowcontrol’ command disables the flow control mechanism. This command is executed to receive and send pause frames individually.

```
switch(config-if)# flowcontrol receive on
switch(config-if)# flowcontrol send on
switch(config-if)#
switch(config-if)# no flowcontrol receive
switch(config-if)# no flowcontrol send on
```

2.1.3. Setting up optional configurations

1. Set up interface description. The description command associates a description with an interface.

```
switch(config-if)# description This is interface 1
switch(config-if)#
```

2. Configure the interface as L2 or L3. By default all interfaces are configured as L3. If an interface is not configured as L3, the routing command can be used to set the interface to L3.

```
switch(config-if)# routing
switch(config-if)#
```

To configure the interface as L2, the no routing command is used.

```
switch(config-if)# no routing
switch(config-if)#
```

3. Set the IP address of the interface. The ip address and ipv6 address commands set the ip address of the interface. These two commands work only if the interface is configured as L3.

```
switch(config-if)# ip address 10.10.10.2/24
switch(config-if)#
switch(config-if)#  ipv6 address 2001:0db8:85a3:0000:0000:8a2e:0370:7334/24
```

To set a secondary ip address, append the ‘secondary’ keyword at the end as shown below:

```
switch(config-if)# ip address 10.10.10.2/24 secondary
switch(config-if)#
switch(config-if)#  ipv6 address 2001:0db8:85a3:0000:0000:8a2e:0370:7334/24 secondar
```

To remove the ipv4/ipv6 interface address, use the no ip address and the no ipv6 address commands.

```
switch(config-if)# no ip address 10.10.10.2/24
switch(config-if)# no ipv6 address 2001:0db8:85a3:0000:0000:8a2e:0370:7334/24
switch(config-if)# no ip address 10.10.10.2/24 secondary
switch(config-if)#  no ipv6 address 2001:0db8:85a3:0000:0000:8a2e:0370:7334/24 secon
```

2.1.4. Viewing interface information

The show interface and show interface brief commands display information about the state and configuration of all the interfaces. The information includes details on speed, mtu, packet counts, and so on.

```
switch# show interface
Interface 45 is down (Administratively down)
Admin state is down
Hardware: Ethernet, MAC Address: 70:72:cf:fd:e7:b4
MTU 1500
Half-duplex
Speed 0 Mb/s
Auto-Negotiation is turned on
Input flow-control is on, output flow-control is on
RX
     0 input packets   0 bytes
     0 input error     0 dropped
     0 short frame     0 overrun
     0 CRC/FCS
TX
     0 output packets   0 bytes
     0 input error      4 dropped
     0 collision
```

```
Interface 36 is down (Administratively down)
Admin state is down
Hardware: Ethernet, MAC Address: 70:72:cf:fd:e7:b4
```

```
MTU 1500
Half-duplex
Speed 0 Mb/s
Auto-Negotiation is turned on
Input flow-control is on, output flow-control is on
RX
     0 input packets   0 bytes
     0 input error     0 dropped
     0 short frame     0 overrun
     0 CRC/FCS
TX
     0 output packets   0 bytes
     0 input error      4 dropped
     0 collision
.........
.........
switch# show interface brief
....................................................................................
Ethernet    VLAN   Type   Mode    Status   Reason                   Speed     Port
Interface                                                           (Mb/s)    Ch#
....................................................................................
45          ..   eth     ..       down    Administratively down    auto      ..
36          ..   eth     ..       down    Administratively down    auto      ..
9           ..   eth     ..       down    Administratively down    auto      ..
...............
...............
```

To view information for a particular interface use the show interface interface or show interface interface brief commands, where the interface variable is the name of the interface, such as interface "1" in the following example.

```
switch# show interface 1
Interface 1 is up
Admin state is up
Hardware: Ethernet, MAC Address: 70:72:cf:fd:e7:b4
MTU 1500
Full-duplex
Speed 1000 Mb/s
Auto-Negotiation is turned on
Input flow-control is off, output flow-control is off
RXc
     50 input packets     14462 bytes
     0 input error        7 dropped
     0 short frame        0 overrun
     0 CRC/FCS
TX
     213 output packets  34506 bytes
     0 input error       4 dropped
     0 collision
switch# show interface 1 brief
....................................................................................
Ethernet    VLAN   Type   Mode    Status     Reason                Speed     Port
```

```
Interface                                                          (Mb/s)    Ch#
....................................................................................
 1          ..    eth     ..      down    Administratively down    auto     ..
```

2.1.5. Viewing snapshot of active configurations

The show running-config interface and show running-config interface interface commands are used to see a snapshot of active configurations for all interfaces and if used with the interface name (interface), active configurations for a particular interface are displayed.

```
switch# show running-config interface
Interface 2
  no shutdown
  speed 40000
  autonegotiation on
  exit
Interface 1
  no shutdown
  exit
.............
.............
switch# show running-config interface 2
Interface 2
  no shutdown
  speed 40000
  autonegotiation on
  exit
```

2.1.6. Troubleshooting the configuration

Condition

Unable to set the ipv4/ipv6 address even after enabling the interface.

Cause

The interface may be configured as an L2.

Remedy

Configure the interface as an L3 using the routing command. See the Command Reference for more information.

2.2. LACP function

2.2.1. Overview

The Link Aggregation Control Protocol (LACP) is one method of bundling several physical interfaces to form one logical interface. LACP exchanges are made between actors and partners. An actor is the local interface in an LACP exchange. A partner is the remote interface in an LACP exchange. LACP is defined in IEEE 802.3ad, Aggregation of Multiple Link Segments.

- In dynamic mode, local Link Aggregation Groups (LAGs) are aware of partner switch LAGs. Interfaces configured as dynamic LAGs are designated as active or passive. — Active interfaces initiate LACP negotiations by sending LACP PDUs when forming a channel with an interface on the remote switch. — Passive interfaces participate in LACP negotiations initiated by remote switch, but are not allowed to initiate such negotiations.

- In static mode, switch LAGs are created without the awareness of their partner switch LAGs. Packets may drop when LAG static aggregate configurations differ between switches. The switch aggregates static links without LACP negotiation.

2.2.2. Prerequisites

All the switch interfaces (at least the interfaces that are connected to other devices) must be administratively up.

2.2.3. Configuring LACP

Creating and adding interfaces to LAG

1. Configure the terminal to change the vtysh context to config context with the following commands:

   ```
   switch# configure terminal
   switch(config)#
   ```

2. Create a LAG with the following command:

   ```
   switch(config)# interface lag 100
   switch(config-lag-if)#
   ```

   After creating the LAG, the CLI drops into LAG interface context and allows you to configure specific LAG parameters.

3. Add interfaces to LAG. A maximum of eight physical interfaces can be added to a LAG. Configure the terminal to change the context to interface context and then add the interface to LAG.

   ```
   switch(config)# interface 1
   switch(config-if)# lag 100
   switch(config-if)#
   ```

2.2.4. Removing interfaces and deleting LAG

1. Configure the terminal to change the vtysh context to config context with the following commands:

   ```
   switch# configure terminal
   switch(config)#
   ```

2. Delete the LAG with the following command:

   ```
   switch(config)# no interface lag 100
   ```

   After deleting the LAG, the interfaces associated with the LAG behave as L3 interfaces.

Remove interfaces from the LAG.

```
switch(config)# interface 1
switch(config-if)# no lag 100
switch(config-if)#
```

2.2.5. Setting up LACP global parameters

Setting the LACP system priority.

```
switch(config)# lacp system-priority 100
switch(config)#
```

The no lacp system-priority commands reverts the LACP system-priority to its default value of 65534.

```
switch(config)# no lacp system-priority
switch(config)#
```

In LACP negotiations, link status decisions are made by the system with the numerically lower priority.

2.2.6. Setting up LAG parameters

1. Setting the LACP mode. LACP mode allows values such as active, passive and off. The default value is off. The following example displays how to set the LACP mode commands to active, passive, or off.

   ```
   switch(config-lag-if)# lacp mode active
   switch(config-lag-if)# lacp mode passive
   switch(config-lag-if)# no lacp mode {active / passive}
   ```

2. Setting the hash type. The Hash type takes the value of l2-src-dst, l3-src-dst or l4-src-dst to control the selection of a interface in a group of aggregate interfaces. This Hash type value helps transmit a frame. The default hash type is l3-src-dst.

   ```
   switch(config-lag-if)# hash l2-src-dst
   ```

3. Setting the LACP rate. LACP rate takes values slow and fast. By default slow. When configured to be fast, LACP heartbeats are requested at a rate of once per second causing connectivity is-

sues to be detected quickly. In slow mode, heartbeats are requested at a rate of once every 30 seconds.

```
switch(config-lag-if)# lacp rate fast
no form of 'lacp rate fast' sets the rate to slow.
switch(config-lag-if)# no lacp rate fast
```

4. Setting the LACP fallback. LACP fallback is used to determine the behavior of a LAG using LACP to negotiate when there is no partner. When the fallback is disabled, which is the default, LAG blocks all its members until it can negotiate with a partner. When fallback is enabled, one or more interfaces are not blocked when there is no partner, depending on the fallback mode, priority (default) or all_active. When priority mode is set, the interface with the higher LACP port-priority is not blocked. When all_active mode is set, none of the LAG interfaces are blocked. LACP fallback timeout is a value in seconds used to determine the time during which fallback will be active. Its default value is zero, meaning that fallback will be active until a partner is detected. It can be configured to be any value between 1 and 900 seconds.

```
switch(config-lag-if)# lacp fallback
switch(config-lag-if)# no lacp fallback
switch(config-lag-if)# lacp fallback mode all_active
switch(config-lag-if)# lacp fallback mode priority
no form of 'lacp fallback mode all_active' sets fallback mode to priority.
switch(config-lag-if)# no lacp fallback mode all_active
```

```
switch(config-lag-if)# lacp fallback timeout 500
no form of 'lacp fallback timeout' sets fallback timeout to zero.
switch(config-lag-if)# no lacp fallback timeout 500
```

You can use show lacp interface to find out if an interface is being unblocked because of the fallback operation. If the interface state is collecting, distributing and has default neighbor state (CDE), it means that the interface is unblocked by fallback.

2.2.7. Setting up interface LACP parameters

1. Setting the LACP port-id. The LACP port-id is used in LACP negotiations to identify individual ports participating in aggregation. The LACP port-id takes values in the range of 1 to 65535.

```
switch(config-if)# lacp port-id 100
```

2. Setting the LACP port-priority. The LACP port-priority is used in LACP negotiations. In LACP negotiations interfaces with numerically lower priorities are preferred for aggregation. The LACP port-priority takes values in the range of 1 to 65535.

```
switch(config-if)# lacp port-priority 100
```

2.2.8. Viewing LACP global information

The show lacp configuration command displays global LACP configuration information.

```
switch# show lacp configuration
System-id       : 70:72:cf:ef:fc:d9
System-priority : 65534
```

2.2.9. Viewing LACP aggregate information

The show lacp aggregates command displays information about all LACP aggregates.

```
switch# show lacp aggregates
Aggregate-name          : lag100
Aggregated-interfaces   :
Heartbeat rate          : slow
Fallback                : false
Fallback mode           : all_active
Fallback timeout        : 50
Hash                    : l3-src-dst
Aggregate mode          : off
Aggregate-name          : lag200
Aggregated-interfaces   :
Heartbeat rate          : slow
Fallback                : false
Fallback mode           : priority
Fallback timeout        : 0
Hash                    : l3-src-dst
Aggregate mode          : off
```

The show lacp aggregates [lag-name] command displays information about specified LAG.

```
switch# show lacp aggregates lag100
Aggregate-name          : lag100
Aggregated-interfaces   :
Heartbeat rate          : slow
Fallback                : false
Fallback mode           : all_active
Fallback timeout        : 50
Hash                    : l3-src-dst
Aggregate mode          : off
```

2.2.10. Viewing LACP interface details

The show lacp interfaces command displays LACP interface configuration.

```
switch# show lacp interfaces
State abbreviations :
A - Active        P - Passive      F - Aggregable I - Individual
S - Short-timeout L - Long-timeout N - InSync     O - OutofSync
C - Collecting    D - Distributing
X - State m/c expired              E - Default neighbor state
```

```
Actor details of all interfaces:
------------------------------------------------------------------------------
Intf Aggregate Port    Port     Key  State   System-id         System   Aggr
     name      id      Priority                                Priority Key
------------------------------------------------------------------------------
3    lag200    69      1        200  ALFNCD  70:72:cf:37:a3:5c 20       200
2    lag200    14      1        200  ALFNCD  70:72:cf:37:a3:5c 20       200
```

```
4    lag200    26      1        200  ALFNCD  70:72:cf:37:a3:5c 20       200
1    lag500    17      1        500  ALFNCD  70:72:cf:37:a3:5c 20       500

Partner details of all interfaces:
------------------------------------------------------------------------------
Intf Aggregate Partner Port     Key  State   System-id         System   Aggr
     name      Port-id Priority                                Priority Key
------------------------------------------------------------------------------
3    lag200    69      1        200  PLFNC   70:72:cf:8c:60:a7 65534    200
2    lag200    14      1        200  PLFNC   70:72:cf:8c:60:a7 65534    200
4    lag200    26      1        200  PLFNCD  70:72:cf:8c:60:a7 65534    200
1    lag500    18      1        500  PLFNCD  70:72:cf:8c:60:a7 65534    500
```

2.3. VLAN function

2.3.1. Overview

This guide provides detail for managing and monitoring VLANs on the switch. All the VLAN configurations work in VLAN context. For a VLAN to have a physical existence, it has to be associated with one of the interfaces. All such configurations work in the interface context. VLAN mandates the associated interface to be non-routing interface. When the VLAN is created, by default it is not associated with any interface. By default VLAN 1 exists. To configure this feature, see Setting up the basic configuration.

The main use of VLANs is to provide network segmentation. VLANs also address issues, such as scalability, security and network management.

Access VLAN

An access VLAN specifies the mode of an interface. By default, VLAN 1 is the access VLAN. An access port carries packets on exactly one VLAN specified. Packets egressing on an access port have no 802.1Q header. Any packet with an 802.1Q header with a nonzero VLAN ID that ingresses on an access port is dropped, regardless of whether the VLAN ID in the header is the access port’s VLAN ID.

Trunk VLAN

A trunk port carries packets on one or more VLANs specified. A packet that ingresses on a trunk port is in the VLAN specified in its 802.1Q header, or native VLAN if the packet has no 802.1Q header. A packet that egresses through a trunk port will have an 802.1Q header if it has a nonzero VLAN ID. Any packet that ingresses on a trunk port tagged with a VLAN that the port does not trunk is dropped.

Native VLAN

Untagged ingress packets are destined to the native VLAN. A native-untagged port resembles a native-tagged port, with the exception that a packet that egresses on a native-untagged port in the native VLAN will not have an 802.1Q header.The native VLAN packets are not tagged when sent out on the trunk links. A native-tagged port resembles a trunk port, with the exception that a packet without an 802.1Q header that ingresses on a native-tagged port is in the native VLAN. Native VLAN packets are tagged when egresses on native-tagged port.

Internal VLAN

A port is configured by default as routed port. When a port is created as routed port an internal VLAN is allocated automatically by system. User cannot configure the internal VLAN. This VLAN information is used internally by the system, This VLAN could be used for L3 interface, sflow, etc. The internal VLAN range is 1024-4096 in ascending order by default.

2.3.2. Setting up the basic configuration

1. Create a VLAN. The vlan <vlanid> command creates a VLAN with a given ID and changes the vtysh context to VLAN. If the VLAN already exists, it then changes the context to VLAN. The

vlanid in the command depicts the name of the VLAN, which is replaced with VLAN *12* in the following example:

```
switch# configure terminal
switch(config)# vlan 12
switch(config-vlan)#
```

The no vlan ID command deletes the VLAN.

```
switch(config)# no vlan 12
switch(config)
```

2. Enable the VLAN. The no shutdown command enables a particular VLAN. Once the VLAN is enabled all the configurations take effect.

```
switch(config-vlan)#no shutdown
switch(config-vlan)#
```

The shutdown command disables a particular VLAN.

```
switch(config-vlan)#shutdown
switch(config-vlan)#
```

3. Add VLAN access to the interface or LAG interface. The vlan access ID command adds an access VLAN to the interface. If the interface is already associated with an access VLAN, then this command overrides the previous configuration. Only one access VLAN can be associated with the interface.

```
switch# config terminal
switch(config)# interface 2
switch(config-if)#no routing
switch(config-if)#vlan access 20
```

The no vlan access command removes the access VLAN from the interface.

```
switch(config-if)#no vlan access
```

4. Adding trunk native VLAN. The vlan trunk native ID command adds a trunk native VLAN to the interface or LAG interface. With this configuration on the interface, all the untagged packets are allowed in the native VLAN.

```
switch# config terminal
switch(config)# interface 21
switch(config-if)#no routing
switch(config-if)#vlan trunk native 1
```

```
switch# config terminal
switch(config)#interface lag 21
switch(config-lag-if)#no routing
switch(config-lag-if)#vlan trunk native 1
```

The *no vlan trunk native* command removes trunk native VLAN from the interface.

```
switch(config-if)#no vlan trunk native
switch(config-lag-if)#no vlan trunk native
```

5. Adding trunk VLAN. The vlan trunk allowed ID command lets you specify the VLAN allowed in the trunk. Multiple VLANs can be allowed on a trunk.

```
switch# config terminal
switch(config)# interface 21
switch(config-if)#no routing
switch(config-if)#vlan trunk allowed 1
```

```
switch# config terminal
switch(config)#interface lag 21
switch(config-lag-if)#no routing
switch(config-lag-if)#vlan trunk allowed 1
```

   The no vlan trunk native command removes the trunk native VLAN specified by ID from the trunk allowed list.

```
switch(config-if)#no vlan trunk allowed 1
switch(config-lag-if)#no vlan trunk allowed 1
```

6. Add tagging on a native VLAN. The vlan trunk native tag command enables tagging on native VLANs.

```
switch# config terminal
switch(config)# interface 21
switch(config-if)#no routing
switch(config-if)#vlan trunk native 1
switch(config-if)#vlan trunk native tag
```

```
switch# config terminal
switch(config)# interface lag 21
switch(config-lag-if)#no routing
switch(config-lag-if)#vlan trunk native 1
switch(config-lag-if)#vlan trunk native tag
```

   The no vlan trunk native tag command disables tagging on native VLANs.

```
switch(config-if)#vlan trunk native tag
switch(config-lag-if)#vlan trunk native tag
```

2.3.3. Verifying the configuration

1. Viewing a VLAN summary. The show vlan summary displays a VLAN summary. The following summary displays a list of configured VLANs:

```
switch# show running-config
Current configuration:
!
vlan 3003
vlan 1
   no shutdown
vlan 1212
```

```
   no shutdown
vlan 33
   no shutdown
vlan 2
  no shutdown
interface bridge_normal
 no routing
```

Though VLAN 1 is configured by default, it is shown in running configuration to ensure that when a switch boots up all ports belong to the default VLAN. This is an exception.

```
switch# show vlan summary
Number of existing VLANs: 5
```

2. Viewing VLAN detailed information. The show vlan shows detailed VLAN configurations. The show vlan ID shows a detailed configuration of a specific VLAN for the following configurations:

```
switch#show running-config
Current configuration:
!
vlan 3003
vlan 1
   no shutdown
vlan 1212
   no shutdown
vlan 33
   no shutdown
vlan 2
  no shutdown
interface bridge_normal
 no routing
interface 2
   no routing
   vlan trunk native 1
   vlan trunk allowed 33
interface 1
   no routing
   vlan access 1
```

```
switch#show vlan
----------------------------------------------------------------------------------
VLAN    Name            Status   Reason         Reserved       Interfaces
----------------------------------------------------------------------------------
1       DEFAULT_VLAN_1  up       ok                            1, 2
2       VLAN2           down     no_member_port
33      VLAN33          up       ok                            2
1212    VLAN1212        down     no_member_port
3003    VLAN3003        down     admin_down
```

```
switch#show vlan 33
----------------------------------------------------------------------------------
```

```
VLAN    Name            Status   Reason         Reserved       Interfaces
----------------------------------------------------------------------------------
33      VLAN33          up       ok                            2
```

2.3.4. Configuring internal VLAN range

The vlan internal range start-vlan-id end-vlan-id [ascending | decsending] command sets the range for internal VLANs in ascending or descending order. For every L3 interfaces, there should be one internal VLAN. Whenever a user configures interfaces, one of the VLAN from this range is assigned to the interface.

```
switch(config)# vlan internal range 4093 4094 ascending
switch(config)# interface 1
switch(config-if)# no shutdown
switch(config-if)# interface 2
switch(config-if)# no shutdown
switch(config-if)# do show vlan internal
```

```
Internal VLAN range  : 4093-4094
Internal VLAN policy : ascending
------------------------
Assigned Interfaces:
       VLAN            Interface
       ----            ---------
       4093            1
       4094            2
```

2.3.5. Troubleshooting an internal VLAN

Every L3 interface must have one internal VLAN. When an interface does not have an internal VLAN because of it running short of internal VLANs that can be checked in the "show vrf" output, the interface that does not have an internal VLAN will have a status of error: no_internal_vlan.

```
switch# configure terminal
switch(config)# vlan internal range 4093 4094 ascending
switch(config)# interface 1
switch(config-if)# no shutdown
switch(config-if)# interface 2
switch(config-if)# no shutdown
switch(config-if)# do show vlan internal
```

```
Internal VLAN range  : 4093-4094
Internal VLAN policy : ascending
------------------------
Assigned Interfaces:
       VLAN            Interface
       ----            ---------
       4093            1
       4094            2
switch(config-if)# interface 3
switch(config-if)# no shutdown
switch(config-if)# do show vlan internal
```

```
Internal VLAN range  : 4093-4094
Internal VLAN policy : ascending
------------------------
Assigned Interfaces:
       VLAN            Interface
       ----            ---------
       4093            1
       4094            2
switch(config-if)# do show vrf
VRF Configuration:
------------------
VRF Name : vrf_default
```

```
       Interfaces :     Status :
       -------------------------
       3                error: no_internal_vlan
       2                up
       1                up
switch(config-if)#
```

2.4. MSTP feature

2.4.1. Overview

The MSTP feature is used for preventing loops in a network. Without a spanning tree, having more than one active path between a pair of nodes causes loops in the network, which can result in duplication of messages and lead to a “broadcast storm” that might bring down the network.

- The Multiple Spanning Tree Protocol (MSTP) (802.1s) ensures that only one active path exists between any two nodes in a spanning tree instance.
- A spanning tree instance comprises a unique set of VLANs, and belongs to a specific spanning tree region.
- A region can comprise multiple spanning tree instances (each with a different set of VLANs), and allows one active path among regions in a network.
- Applying VLAN tagging to the ports in a multiple-instance spanning tree network enables blocking of redundant links in one instance while allowing forwarding over the same links for non-redundant use by another instance.

This feature currently works on a physical interface and LAGs.

2.4.2. MSTP Structure

MSTP maps active, separate paths through separate spanning tree instances and between MST regions. Each MST region comprises one or more MSTP switches.

- The common and internal spanning tree (CIST) identifies the regions in a network and administers the CIST root bridge for the network, the root bridge for each region, and the root bridge for each spanning tree instance in each region.
- The CIST root administers the connectivity among the MST regions in a bridged network.
- An MST region comprises the VLANs configured on physically connected MSTP switches. All switches in a given region must be configured with the same VLANs and Multiple Spanning Tree Instances (MSTIs).
- The internal spanning tree (IST) administers the topology within a given MST region. When a switch is configured for MSTP operation, the switch automatically includes all of the static VLANs configured on the switch in a single, active spanning tree topology (instance) within the IST. This is termed the “IST instance”. Any VLANs you subsequently configure on the switch are added to this IST instance. To create separate forwarding paths within a region, group specific VLANs into different multiple spanning tree instances (MSTIs).

2.4.3. Types of multiple spanning tree instances

A network can have several MSTP regions. An MSTP region has multiple spanning tree instances. Each instance defines a single forwarding topology for an exclusive set of VLANs.

- Internal spanning tree instance (IST instance) This is the default spanning tree instance in any MST region. It provides the root switch for the region and comprises all VLANs configured on the switches in the region that are not specifically assigned to multiple spanning tree instances (MSTIs, described below). All VLANs in the IST instance of a region are part of the same, single spanning tree topology, which allows only one forwarding path between any two nodes belonging to any of the VLANs included in the IST instance. All switches in the region must belong to the set of VLANs that comprise the IST instance.

- Multiple spanning tree instance (MSTI) This type of configurable spanning tree instance comprises all static VLANs specifically assign to it, and must include at least one VLAN. The VLAN(s) assigned to an MSTI must initially exist in the IST instance of the same MST region. When a static VLAN is assigned to an MSTI, the switch removes the VLAN from the IST instance. (Thus, a VLAN is assigned to only one MSTI in a given region.) All VLANs in an MSTI operate as part of the same single spanning tree topology.

2.4.4. How MSTP Operates

- In the factory default configuration, spanning tree operation is off. Also, the switch retains its currently configured spanning tree parameter settings when disabled. Thus, if spanning tree is disabled, then later re-enabled, the parameter settings are the same as before spanning tree was disabled.

- All MSTP switches in a given region must be configured with the same VLANs. Also, each MSTP switch within the same region must have the same VLAN-to-instance assignments. (A VLAN can belong to only one instance within any region.) Within a region: — All of the VLANs belonging to a given instance compose a single, active spanning tree topology for that instance. — Each instance operates independently of other regions. — Between regions there is a single, active spanning tree topology.

2.4.5. Terminology

- Common and internal spanning tree (CIST) Comprises all LANs and MSTP regions in a network. The CIST automatically determines the MST regions in a network and defines the root bridge (switch) and designated port for each region. The CIST includes the Common Spanning Tree (CST), the Internal Spanning Tree (IST) within each region, and any multiple spanning-tree instances (MSTIs) in a region.

- Internal spanning tree (IST) Comprises all VLANs within a region that are not assigned to a MSTI configured within the region. All MSTP switches in a region should belong to the IST. In a given region “X”, the IST root switch is the regional root switch and provides information on region “X” to other regions.

- MSTP (Multiple Spanning Tree Protocol) A network supporting MSTP allows multiple spanning tree instances within configured regions, and a single spanning tree among regions.

- MSTP BPDU (Bridge Protocol Data Unit) BPDUs carry region-specific information, such as the region identifier (region name and revision number). If a switch receives an MSTP BPDU with a region identifier that differs from its own, then the port on which that BPDU was received is on the boundary of the region in which the switch resides.

- MSTP bridge In this manual, an MSTP bridge is a switch (or another 802.1s compatible device) configured for MSTP operation.

- MST region A MST region forms a multiple spanning tree domain and is a component of a single spanning tree domain within a network. For switches internal to the MST region: — All switches have identical MST configuration identifiers (region name and revision number). — All switches have identical VLAN assignments to the region’s IST and (optional) MST instances. — One switch functions as the designated bridge (IST root) for the region. — No switch has a point-to-point connection to a bridging device that cannot process MSTP BPDUs.

2.4.6. Operating rules

- All switches in a region must be configured with the same set of VLANs, as well as the same MST configuration name and MST configuration number.
- Within a region, a VLAN can be allocated to either a single MSTI or to the region’s IST instance.
- All switches in a region must have the same VLAN ID to MST instance and VLAN ID to IST instance assignments.
- There is one root MST switch per configured MST instance.
- Within any region, the root switch for the IST instance is also the root switch for the region. Because boundary ports provide the VLAN connectivity between regions, all boundary ports on a region’s root switch should be configured as members of all static VLANs defined in the region.
- There is one root switch for the CIST. Note that the per-port hello-time parameter assignments on the CIST root switch propagate to the ports on downstream switches in the network, and override the hello-time configured on the downstream switch ports.
- Where multiple MST regions exist in a network, there is only one active, physical communication path between any two regions, or between an MST region and an STP or RSTP switch. MSTP blocks any other physical paths as long as the currently active path remains in service.
- Within an MSTI, there is one spanning tree (one physical, communication path) between any two nodes. That is, within an MSTI, there is one instance of spanning tree, regardless of how many VLANs belong to the MSTI. Within an IST instance, there is also one spanning tree across all VLANs belonging to the IST instance.
- An MSTI comprises a unique set of VLANs and forms a single spanning tree instance within the region to which it belongs.
- An MSTI should have at least one VLAN configured to it.
- Removing an MSTI through CLI or REST moves the configured VLANs from MSTI to IST.
- Communication between MST regions uses a single spanning tree.
- If a port on a switch configured for MSTP receives a legacy (STP/802.1D or RSTP/802.1w) BPDU, it automatically operates as a legacy port. In this case, the MSTP switch interoperates with the connected STP or RSTP switch as a separate MST region.
- Within an MST region, there is one logical forwarding topology per instance, and each instance comprises a unique set of VLANs. Where multiple paths exist between a pair of nodes using VLANs belonging to the same instance, all but one of those paths is blocked for that instance. However, if there are different paths in different instances, all such paths are available for traffic. Separate forwarding paths exist through separate spanning tree instances.
- A port can have different states (forwarding or blocking) for different instances (which represent different forwarding paths).

2.4.7. Usage scenarios

2.4.7.1. Setting up scenario 1 basic configuration

Create a three switch topology as specified below.

Physical Topology

```
+---------------------------------------------------------------------------+
|    Region "A": Physical Topology                                          |
|                            +----------------+                             |
|                Link-2      +      SW_1      +      Link-1                 |
|          +-----------------+                +----------------+            |
|          |                 +                +                |            |
|          |                 +----------------+                |            |
|          +                                                   +            |
|   +------+---------+                                +--------+--------+   |
|   |                |            Link-3              +                 |   |
|   |     SW_3       +--------------------------------+      SW_2       |   |
|   |                |                                +                 |   |
|   +----------------+                                +-----------------+   |
|                                                                           |
+---------------------------------------------------------------------------+
```

Configure instance 1 and 2 as specified below:

```
VLANS      | Instance 1 | Instance 2
-----    | ---------- | ---------
10,11,12 | Yes           | No
20,21,22 | No           | Yes
```

The logical and physical topologies resulting from these VLAN/instance groupings result in blocking on different links for different VLANs:

Logical Topology

```
+---------------------------------------------------------------------------+
| Logical topology for Instance-1                                           |
|                          +--------------------+                           |
|             Link-2       |       SW-1         |   Link-1                  |
|        +-----------------+ Root for Instance 1+---------------+           |
|        |                 |  VLANs = 10,11,12  |               |           |
|        |                 +--------------------+               |           |
|        |                                                      |           |
|        |                                                      |           |
| +------+----------+                                  +--------+---------+ |
| |     SW-2        |           Link-3(Blocked)        |     SW-3         | |
| |  Instance-1     +----------------------------------+  Instance-1      | |
| | VLANs = 10,11,12|                                  | VLANS = 10,11,12 | |
| +-----------------+                                  +------------------+ |
|                                                                           |
+---------------------------------------------------------------------------+
+---------------------------------------------------------------------------+
| Logical topology for Instance-2                                           |
|                          +--------------------+                           |
|             Link-2       |       SW-1         |   Link-1(Blocked)         |
|        +-----------------+ Root for Instance 2+---------------+           |
|        |                 |  VLANs = 20,21,22  |               |           |
|        |                 +--------------------+               |           |
|        |                                                      |           |
|        |                                                      |           |
| +------+----------+                                  +--------+---------+ |
| |     SW-2        |           Link-3                 |     SW-3         | |
| |  Instance-2     +----------------------------------+  Instance-2      | |
| | VLANs = 20,21,22|                                  | VLANS = 20,21,22 | |
| +-----------------+                                  +------------------+ |
|                                                                           |
+---------------------------------------------------------------------------+
```

MSTP uses VLANs to create multiple spanning trees in a network, which significantly improves network resource utilization while maintaining a loop-free environment. Thus, where a port belongs to multiple VLANs, it may be dynamically blocked in one spanning tree instance, but forwarding in another instance. This achieves load-balancing across the network while keeping the switch’s CPU load at a moderate level (by aggregating multiple VLANs in a single spanning tree instance).

2.4.7.2. Configure MSTP global parameters

Configure the same region name and revision number in all three switches:

```
spanning-tree config-name mst
spanning-tree config-revision 1
spanning-tree
```

Create two MSTP instances and map the VLANS as shown below:

```
spanning-tree instance 1 vlan 10
spanning-tree instance 1 vlan 11
spanning-tree instance 1 vlan 12
```

```
spanning-tree instance 2 vlan 20
spanning-tree instance 2 vlan 21
spanning-tree instance 2 vlan 22
```

2.4.7.3. Configure MSTP optional parameters

```
spanning-tree max-hops 10
spanning-tree hello-time 8
spanning-tree forward-delay 8
```

2.4.7.4. Verifying scenario 1 configuration

```
MSTP details on Root switch:
```

```
switch# sh spanning-tree mst detail
#### MST0
Vlans mapped:  1-9,11-4095
```

```
Bridge         Address:70:72:cf:03:d3:e9    priority:32768
Root
Regional Root
Operational    Hello time(in seconds): 2  Forward delay(in seconds):15
    Max-age(in seconds):20  txHoldCount(in pps): 6
Configured     Hello time(in seconds): 2  Forward delay(in seconds):15
    Max-age(in seconds):20  txHoldCount(in pps): 6
```

```
Port           Role           State      Cost       Priority   Type
-------------- -------------- ---------- ---------- ---------- ----------
1              Designated     Forwarding 0          128        point_to_point
2              Designated     Forwarding 0          128        point_to_point
```

```
#### MST1
Vlans mapped:  10
Bridge         Address:70:72:cf:03:d3:e9    Priority:32768
Root           Address:70:72:cf:03:d3:e9    Priority:32769
               Port:, Cost:20000, Rem Hops:0
```

```
Port           Role           State      Cost    Priority   Type
-------------- -------------- ---------- ------- ---------- ----------
1              Designated     Forwarding 0       128        point_to_point
2              Designated     Forwarding 0       128        point_to_point
```

```
Port 1
Designated root address            : 70:72:cf:03:d3:e9
Designated regional root address   : 70:72:cf:03:d3:e9
Designated bridge address          : 32768.0.70:72:cf:03:d3:e9
Timers:    Message expires in 1 sec, Forward delay expiry:18, Forward
    transitions:18
Bpdus sent 183, received 4
```

```
Port 2
Designated root address            : 70:72:cf:03:d3:e9
Designated regional root address   : 70:72:cf:03:d3:e9
Designated bridge address          : 32768.0.70:72:cf:03:d3:e9
Timers:    Message expires in 1 sec, Forward delay expiry:18, Forward
    transitions:18
Bpdus sent 183, received 4
```

```
MSTP details on Non-Root switch:
```

```
switch# sh spanning-tree mst detail
#### MST0
Vlans mapped:  1-9,11-4095
Bridge         Address:70:72:cf:e7:25:b1    priority:32768
Operational    Hello time(in seconds): 2  Forward delay(in seconds):15
    Max-age(in seconds):20  txHoldCount(in pps): 6
Configured     Hello time(in seconds): 2  Forward delay(in seconds):15
    Max-age(in seconds):20  txHoldCount(in pps): 6
```

```
Port           Role           State      Cost       Priority   Type
-------------- -------------- ---------- ---------- ---------- ----------
1              Root           Forwarding 2000000    128        point_to_point
```

```
2              Alternate      Blocking   4000000    128        point_to_point
```

```
#### MST1
Vlans mapped:  10
Bridge         Address:70:72:cf:e7:25:b1    Priority:32768
Root           Address:70:72:cf:3b:ea:a4    Priority:32769
               Port:1, Cost:4000000, Rem Hops:20
```

```
Port           Role           State      Cost    Priority   Type
-------------- -------------- ---------- ------- ---------- ----------
1              Root           Forwarding 2000000 128        point_to_point
2              Alternate      Blocking   2000000 128        point_to_point
```

```
Port 1
Designated root address            : 70:72:cf:3b:ea:a4
Designated regional root address   : 70:72:cf:e7:25:b1
Designated bridge address          : 32768.0.70:72:cf:e7:25:b1
Timers:    Message expires in 0 sec, Forward delay expiry:0, Forward
    transitions:0
Bpdus sent 4, received 1853
```

```
Port 2
Designated root address            : 70:72:cf:3b:ea:a4
Designated regional root address   : 70:72:cf:e7:25:b1
Designated bridge address          : 32768.0.70:72:cf:e7:25:b1
Timers:    Message expires in 0 sec, Forward delay expiry:0, Forward
    transitions:0
Bpdus sent 5, received 1854
```

```
switch#
```

2.4.8. Troubleshooting

Duplicate packets on a VLAN, or packets not arriving on a LAN at all.

- The allocation of VLANs to MSTIs may not be identical among all switches in a region.
- Check the current instance to VLAN mapping by show spanning-tree mst-config:

```
switch# sh spanning-tree mst-config
MST configuration information
MST config ID        : mst2
MST config revision  : 2
MST config digest    : 870555C957F1B44530B7D56FD4716ADF
Number of instances  : 1
```

```
Instance ID     Member VLANs
--------------- ----------------------------------
1               10,11,12
2               20,21,22
switch#
```

A switch intended to operate within a region does not receive traffic from other switches in the region.

- An MSTP switch intended for a particular region may not have the same configuration name or region revision number as the other switches intended for the same region.
- The set of VLANs configured on the switch may not match the set of VLANs configured on other switches in the intended region.
- Check the config-name and revision using the show running-config spanning-tree command or the show spanning-tree mst-config command.

MSTP port roles are always blocked.

- Check that the admin status for the corresponding ports are up using the show interface command:

```
switch# show interface 1
Interface 1 is up
Admin state is up
Hardware: Ethernet, MAC Address: 70:72:cf:3b:ea:a4
MTU 1500
Full-duplex
qos trust none
qos queue-profile default
qos schedule-profile default
Speed 1000 Mb/s
Auto-Negotiation is turned on
Input flow-control is off, output flow-control is off
RX
            5037 input packets         548385 bytes
               0 input error                0 dropped
               0 CRC/FCS
TX
          135373 output packets      17976260 bytes
               0 input error                0 dropped
               0 collision
switch#
```

The network is not stable, convergence restarts after every few seconds.

- Check the BPDU statistics using the show spanning-tree detail command.
- Root bridge should not receive any non-root bridge BPDU after convergence is completed.
- The Number of topology changes value should not increase rapidly if no network changes are happening.
- On non-root bridges BPDU received should not increase, and on the other way BPDU sent count should not increase.

```
switch# sh spanning-tree detail
MST0
Spanning tree status: Enabled
Root ID    Priority   : 32768
MAC-Address: 70:72:cf:3b:ea:a4
This bridge is the root
Hello time(in seconds):2  Max Age(in seconds):20  Forward Delay(in seconds):15
```

```
Bridge ID  Priority  : 32768
MAC-Address: 70:72:cf:3b:ea:a4
Hello time(in seconds):2  Max Age(in seconds):20  Forward Delay(in seconds):15
```

```
Port         Role           State      Cost    Priority   Type
------------ -------------- ---------- ------- ---------- ----------
1            Designated     Forwarding 0       128        point_to_point
2            Designated     Forwarding 0       128        point_to_point
```

```
Topology change flag          : True
Number of topology changes    : 2
Last topology change occurred : 247012 seconds ago
Timers:    Hello expiry  1 , Forward delay expiry 0
```

```
Port 1
Designated root has priority               :32768 Address: 70:72:cf:3b:ea:a4
Designated bridge has priority             :32768 Address: 70:72:cf:3b:ea:a4
Designated port                            :1
Number of transitions to forwarding state  : 0
Bpdus sent 123508, received 2
```

```
Port 2
Designated root has priority               :32768 Address: 70:72:cf:3b:ea:a4
Designated bridge has priority             :32768 Address: 70:72:cf:3b:ea:a4
Designated port                            :2
Number of transitions to forwarding state  : 0
Bpdus sent 123508, received 2
```

- Root switch should not get multiple topology change events from other non-root bridges. Check this using the show events category mstp command.

```
2016-06-04:12:06:51.683147|ops-stpd|23012|LOG_INFO|CIST - Topology Change
generated on port 1 going in to forwarding
2016-06-04:12:06:51.686827|ops-stpd|23012|LOG_INFO|CIST - Topology Change
generated on port 2 going in to forwarding
2016-06-04:12:06:51.689134|ops-stpd|23001|LOG_INFO|MSTP Enabled
2016-06-04:12:06:51.700332|ops-stpd|23011|LOG_INFO|Topology Change received
on port 1 for CIST from source: 48:0f:cf:af:d3:93
2016-06-04:12:06:51.705785|ops-stpd|23011|LOG_INFO|Topology Change received
on port 2 for CIST from source: 48:0f:cf:af:65:b9
2016-06-04:12:06:52.987110|ops-stpd|23011|LOG_INFO|Topology Change received
on port 2 for CIST from source: 48:0f:cf:af:65:b9
2016-06-04:12:09:07.348952|ops-stpd|23011|LOG_INFO|Topology Change received
on port 2 for CIST from source: 48:0f:cf:af:65:b9
2016-06-04:12:09:08.986934|ops-stpd|23011|LOG_INFO|Topology Change received
on port 2 for CIST from source: 48:0f:cf:af:65:b9
2016-06-04:12:09:11.707898|ops-stpd|23011|LOG_INFO|Topology Change received
on port 2 for CIST from source: 48:0f:cf:af:65:b9
2016-06-04:12:09:11.725218|ops-stpd|23011|LOG_INFO|Topology Change received
on port 1 for CIST from source: 48:0f:cf:af:d3:93
2016-06-04:12:09:12.986157|ops-stpd|23011|LOG_INFO|Topology Change received
on port 2 for CIST from source: 48:0f:cf:af:65:b9
```

```
2016-06-04:12:09:13.641048|ops-stpd|23011|LOG_INFO|Topology Change received
on port 1 for CIST from source: 48:0f:cf:af:d3:93
2016-06-04:12:09:17.044583|ops-stpd|23011|LOG_INFO|Topology Change received
on port 1 for CIST from source: 48:0f:cf:af:d3:93
2016-06-04:12:09:17.474762|ops-stpd|23011|LOG_INFO|Topology Change received
on port 2 for CIST from source: 48:0f:cf:af:65:b9
2016-06-04:12:09:18.986615|ops-stpd|23011|LOG_INFO|Topology Change received
on port 2 for CIST from source: 48:0f:cf:af:65:b9
2016-06-04:12:09:23.577870|ops-stpd|23012|LOG_INFO|CIST - Topology Change
generated on port 1 going in to forwarding
2016-06-04:12:09:23.578935|ops-stpd|23012|LOG_INFO|MSTI 1 - Topology Change
generated on port 1 going in to forwarding
2016-06-04:12:09:23.582833|ops-stpd|23012|LOG_INFO|CIST - Topology Change
generated on port 2 going in to forwarding
2016-06-04:12:09:23.583842|ops-stpd|23012|LOG_INFO|MSTI 1 - Topology Change
generated on port 2 going in to forwarding
2016-06-04:12:09:23.586780|ops-stpd|23001|LOG_INFO|MSTP Enabled
```

Additional tips for troubleshooting

Data can be retrieved from the MSTP daemon by running the diag-dump mstp basic command.

```
switch# diag-dump mstp basic
=========================================================================
[Start] Feature mstp Time : Mon Jun 20 12:36:05 2016
```

```
=========================================================================
-------------------------------------------------------------------------
[Start] Daemon ops-stpd
-------------------------------------------------------------------------
MSTP CIST Config OVSDB info:
MSTP VLANs:
MSTP CIST Priority 8
MSTP CIST Hello Time 2
MSTP CIST Forward Delay 15
MSTP CIST Max Age 20
MSTP CIST Max Hop Count 20
MSTP CIST Tx Hold Count 6
mstpEnabled       : Yes
valid             : Yes
cistRootPortID    : port#=0, priority=0
CistBridgeTimes   : {fwdDelay=15 maxAge=20 messageAge=0 hops=20}
cistRootTimes     : {fwdDelay=15 maxAge=20 messageAge=0 hops=20}
cistRootHelloTime : 0
BridgeIdentifier  : {mac=70:72:cf:18:3e:0c priority=32768 sysID=0}
CistBridgePriority:
        rootID      {mac=70:72:cf:18:3e:0c priority=32768 sysID=0}
        extRootPathCost=0
        rgnRootID   {mac=70:72:cf:18:3e:0c priority=32768 sysID=0}
        intRootPathCost=0
        dsnBridgeID {mac=70:72:cf:18:3e:0c priority=32768 sysID=0}
        dsnPortID=(0;0)
cistRootPriority  :
        rootID      {mac=70:72:cf:18:3e:0c priority=32768 sysID=0}
        extRootPathCost=0
        rgnRootID   {mac=70:72:cf:18:3e:0c priority=32768 sysID=0}
        intRootPathCost=0
        dsnBridgeID {mac=70:72:cf:18:3e:0c priority=32768 sysID=0}
        dsnPortID=(0;0)
SM states         : PRS=ROLE_SELECTION
TC Trap Control   : false
MSTP CIST Lport : 2
MSTP CIST PORT Priority : 8
MSTP CIST PORT Admin Path Cost : 0
MSTP CIST PORT Admin Edge port : 0
MSTP CIST PORT BPDUS RX Enable : 0
MSTP CIST PORT BPDUS TX Enable : 0
MSTP CIST PORT Restricted Port Role : 0
MSTP CIST PORT Restricted Port Tcn : 0
MSTP CIST PORT BPDU Guard : 0
MSTP CIST PORT LOOP Guard : 0
MSTP CIST PORT ROOT Guard : 0
MSTP CIST PORT BPDU Filter : 0
SM Timers     : fdWhile=0 rrWhile=0 rbWhile=0 tcWhile=0 rcvdInfoWhile=0
Perf Params   : InternalPortPathCost=2000000, useCfgPathCost=F
Per-Port Vars :
  portId=(128;2) infoIs=MINE rcvdInfo=INFERIOR_ROOT_ALT
  role=DESIGNATED selectedRole=DESIGNATED
  cistDesignatedTimes={fwdDelay=15 maxAge=20 messageAge=0 hops=20}
  cistMsgTimes       ={fwdDelay=15 maxAge=20 messageAge=0 hops=19 helloTime=2}
  cistPortTimes      ={fwdDelay=15 maxAge=20 messageAge=0 hops=20 helloTime=2}
  cistDesignatedPriority=
     {rootID     =(70:72:cf:18:3e:0c;32768;0) : extRootPathCost=0 :
      rgnRootID  =(70:72:cf:18:3e:0c;32768;0) : intRootPathCost=0 :
      dsnBridgeID=(70:72:cf:18:3e:0c;32768;0} : dsnPortID=(128;2)}
  cistMsgPriority=
     {rootID     =(70:72:cf:18:3e:0c;32768;0} : extRootPathCost=0 :
      rgnRootID  =(70:72:cf:18:3e:0c;32768;0) : intRootPathCost=20000 :
      dsnBridgeID=(70:72:cf:5f:3e:25;32768;0) : dsnPortID=(128;1)}
  cistPortPriority=
     {rootID     =(70:72:cf:18:3e:0c;32768;0} : extRootPathCost=0
      rgnRootID  =(70:72:cf:18:3e:0c;32768;0) : intRootPathCost=0
      dsnBridgeID=(70:72:cf:18:3e:0c;32768;0) : dsnPortID=(128;2)}
Flags    : FWD=1 FWDI=1 LRN=1  LRNI=1  PRPSD=0 PRPSI=0 RROOT=0 RSELT=0  SELTD=1
        AGR=1 AGRD=1 SYNC=0 SYNCD=1 TCPRP=0 UPDT=0  RCVTC=0 RCVMSG=0 CMSTR=0
SM states: PIM=CURRENT       PRT=DSGN_PORT    PST=FORWARDING TCM=ACTIVE
MSTP CIST Lport : 1
MSTP CIST PORT Priority : 8
MSTP CIST PORT Admin Path Cost : 0
MSTP CIST PORT Admin Edge port : 0
MSTP CIST PORT BPDUS RX Enable : 0
MSTP CIST PORT BPDUS TX Enable : 0
MSTP CIST PORT Restricted Port Role : 0
MSTP CIST PORT Restricted Port Tcn : 0
MSTP CIST PORT BPDU Guard : 0
MSTP CIST PORT LOOP Guard : 0
MSTP CIST PORT ROOT Guard : 0
MSTP CIST PORT BPDU Filter : 0
SM Timers     : fdWhile=0 rrWhile=0 rbWhile=0 tcWhile=0 rcvdInfoWhile=0
Perf Params   : InternalPortPathCost=2000000, useCfgPathCost=F
Per-Port Vars :
   portId=(128;1) infoIs=MINE rcvdInfo=INFERIOR_ROOT_ALT
   role=DESIGNATED selectedRole=DESIGNATED
   cistDesignatedTimes={fwdDelay=15 maxAge=20 messageAge=0 hops=20}
   cistMsgTimes       ={fwdDelay=15 maxAge=20 messageAge=0 hops=19 helloTime=2}
   cistPortTimes      ={fwdDelay=15 maxAge=20 messageAge=0 hops=20 helloTime=2}
   cistDesignatedPriority=
      {rootID     =(70:72:cf:18:3e:0c;32768;0) : extRootPathCost=0 :
       rgnRootID  =(70:72:cf:18:3e:0c;32768;0) : intRootPathCost=0 :
       dsnBridgeID=(70:72:cf:18:3e:0c;32768;0} : dsnPortID=(128;1)}
   cistMsgPriority=
      {rootID     =(70:72:cf:18:3e:0c;32768;0} : extRootPathCost=0 :
       rgnRootID  =(70:72:cf:18:3e:0c;32768;0) : intRootPathCost=2000000 :
       dsnBridgeID=(70:72:cf:36:0e:63;32768;0) : dsnPortID=(128;1)}
   cistPortPriority=
      {rootID     =(70:72:cf:18:3e:0c;32768;0} : extRootPathCost=0
       rgnRootID  =(70:72:cf:18:3e:0c;32768;0) : intRootPathCost=0
       dsnBridgeID=(70:72:cf:18:3e:0c;32768;0) : dsnPortID=(128;1)}
Flags    : FWD=1 FWDI=1 LRN=1  LRNI=1  PRPSD=0 PRPSI=0 RROOT=0 RSELT=0  SELTD=1
           AGR=1 AGRD=1 SYNC=0 SYNCD=1 TCPRP=0 UPDT=0  RCVTC=0 RCVMSG=0 CMSTR=0
SM states: PIM=CURRENT       PRT=DSGN_PORT    PST=FORWARDING TCM=ACTIVE
MSTP MSTI Config OVSDB info:
MSTP MSTI VLANs: 10
MSTP MSTI Priority 8
```

```
mstpEnabled       : Yes
valid             : Yes
vlanGroupNum      : 0
mstiRootPortID    : port#=0, priority=0
MstiBridgeTimes   : {hops=20}
mstiRootTimes     : {hops=20}
BridgeIdentifier  : {mac=70:72:cf:18:3e:0c priority=32768 sysID=1}
MstiBridgePriority:
        rgnRootID   {mac=70:72:cf:18:3e:0c priority=32768 sysID=1}
        intRootPathCost=0
        dsnBridgeID {mac=70:72:cf:18:3e:0c priority=32768 sysID=1}
        dsnPortID=(0;0)
mstiRootPriority  :
        rgnRootID   {mac=70:72:cf:18:3e:0c priority=32768 sysID=1}
        intRootPathCost=0
        dsnBridgeID {mac=70:72:cf:18:3e:0c priority=32768 sysID=1}
        dsnPortID=(0;0)
SM states         : PRS=ROLE_SELECTION
```

```
Total BPDU Filters activated: 0
TC Trap Control   : false
```

```
MSTP MSTI Lport : 2
MSTP CIST PORT Priority : 8
MSTP CIST PORT Admin Path Cost : 0
```

```
SM Timers     : fdWhile=0 rrWhile=0 rbWhile=0 tcWhile=0 rcvdInfoWhile=0
Perf Params   : InternalPortPathCost=2000000, useCfgPathCost=F
Per-Port Vars :
    portId=(128;2) infoIs=MINE rcvdInfo=INFERIOR_ROOT_ALT
    role=DESIGNATED selectedRole=DESIGNATED
    mstiDesignatedTimes={hops=20}
    mstiMsgTimes       ={hops=19}
    mstiPortTimes      ={hops=20}
    mstiDesignatedPriority=
       {rgnRootID  =(70:72:cf:18:3e:0c;32768;1) : intRootPathCost=0 :
        dsnBridgeID=(70:72:cf:18:3e:0c;32768;1} : dsnPortID=(128;2)}
    mstiMsgPriority=
       {rgnRootID  =(70:72:cf:18:3e:0c;32768;1) : intRootPathCost=20000 :
        dsnBridgeID=(70:72:cf:5f:3e:25;32768;1} : dsnPortID=(128;1)}
    mstiPortPriority=
       {rgnRootID  =(70:72:cf:18:3e:0c;32768;1) : intRootPathCost=0 :
        dsnBridgeID=(70:72:cf:18:3e:0c;32768;1} : dsnPortID=(128;2)}
Flags    : FWD=1 FWDI=1 LRN=1  LRNI=1  PRPSD=0 PRPSI=0 RROOT=0 RSELT=0 SELTD=1
           AGR=1 AGRD=1 SYNC=0 SYNCD=1 TCPRP=0 UPDT=0  RCVTC=0 RCVMSG=0
           MSTR=0       MSTRD=0
SM states: PIM=CURRENT       PRT=DSGN_PORT    PST=FORWARDING TCM=ACTIVE
```

```
MSTP MSTI Lport : 1
MSTP CIST PORT Priority : 8
MSTP CIST PORT Admin Path Cost : 0
```

```
SM Timers     : fdWhile=0 rrWhile=0 rbWhile=0 tcWhile=0 rcvdInfoWhile=0
Perf Params   : InternalPortPathCost=2000000, useCfgPathCost=F
Per-Port Vars :
   portId=(128;1) infoIs=MINE rcvdInfo=INFERIOR_ROOT_ALT
   role=DESIGNATED selectedRole=DESIGNATED
   mstiDesignatedTimes={hops=20}
   mstiMsgTimes       ={hops=19}
   mstiPortTimes      ={hops=20}
   mstiDesignatedPriority=
      {rgnRootID  =(70:72:cf:18:3e:0c;32768;1) : intRootPathCost=0 :
       dsnBridgeID=(70:72:cf:18:3e:0c;32768;1} : dsnPortID=(128;1)}
   mstiMsgPriority=
      {rgnRootID  =(70:72:cf:18:3e:0c;32768;1) : intRootPathCost=2000000 :
       dsnBridgeID=(70:72:cf:36:0e:63;32768;1} : dsnPortID=(128;1)}
   mstiPortPriority=
      {rgnRootID  =(70:72:cf:18:3e:0c;32768;1) : intRootPathCost=0 :
       dsnBridgeID=(70:72:cf:18:3e:0c;32768;1} : dsnPortID=(128;1)}
Flags    : FWD=1 FWDI=1 LRN=1  LRNI=1  PRPSD=0 PRPSI=0 RROOT=0 RSELT=0 SELTD=1
           AGR=1 AGRD=1 SYNC=0 SYNCD=1 TCPRP=0 UPDT=0  RCVTC=0 RCVMSG=0
           MSTR=0       MSTRD=0
SM states: PIM=CURRENT       PRT=DSGN_PORT    PST=FORWARDING TCM=ACTIVE
```

```
------------------------------------------------------------------------
```

```
[End] Daemon ops-stpd
-------------------------------------------------------------------------
=========================================================================
[End] Feature mstp
=========================================================================
Diagnostic dump captured for feature mstp
switch#
```

Information to debug an MSTP issue can be obtained by running the show tech mstp command:

```
switch# show tech mstp
====================================================
Show Tech executed on Mon Jun 20 05:47:40 2016
====================================================
====================================================
[Begin] Feature mstp
====================================================
```

```
*********************************
Command : show spanning-tree detail
*********************************
MST0
  Spanning tree status: Enabled
  Root ID    Priority   : 32768
             MAC-Address: 70:72:cf:3b:ea:a4
             This bridge is the root
             Hello time(in seconds):2  Max Age(in seconds):20  Forward Delay(in seco
```

```
Bridge ID  Priority  : 32768
           MAC-Address: 70:72:cf:3b:ea:a4
           Hello time(in seconds):2  Max Age(in seconds):20  Forward Delay(in second
```

```
Port         Role           State      Cost    Priority   Type
------------ -------------- ---------- ------- ---------- ----------
1            Designated     Forwarding 0       128        point_to_point
2            Designated     Forwarding 0       128        point_to_point
```

```
Topology change flag          : True
Number of topology changes    : 2
Last topology change occurred : 248988 seconds ago
Timers:    Hello expiry  1 , Forward delay expiry 0
```

```
Port 1
Designated root has priority               :32768 Address: 70:72:cf:3b:ea:a4
Designated bridge has priority             :32768 Address: 70:72:cf:3b:ea:a4
Designated port                            :1
Number of transitions to forwarding state  : 0
Bpdus sent 124496, received 2
```

```
Port 2
Designated root has priority               :32768 Address: 70:72:cf:3b:ea:a4
Designated bridge has priority             :32768 Address: 70:72:cf:3b:ea:a4
Designated port                            :2
```

```
Number of transitions to forwarding state  : 0
Bpdus sent 124496, received 2
```

```
*********************************
Command : show spanning-tree mst-config
*********************************
MST configuration information
   MST config ID        : mst2
   MST config revision  : 2
   MST config digest    : 870555C957F1B44530B7D56FD4716ADF
   Number of instances  : 1
```

```
Instance ID     Member VLANs
--------------- ----------------------------------
1               10
```

```
*********************************
Command : show spanning-tree mst
*********************************
#### MST0
Vlans mapped:  1-9,11-4095
Bridge         Address:70:72:cf:3b:ea:a4    priority:32768
Root
Regional Root
Operational    Hello time(in seconds): 2  Forward delay(in seconds):15
Max-age(in seconds):20  txHoldCount(in pps): 6
Configured     Hello time(in seconds): 2  Forward delay(in seconds):15
Max-age(in seconds):20  txHoldCount(in pps): 6
```

```
Port           Role           State      Cost       Priority   Type
-------------- -------------- ---------- ---------- ---------- ----------
1              Designated     Forwarding 0          128        point_to_point
2              Designated     Forwarding 0          128        point_to_point
```

```
#### MST1
Vlans mapped:  10
Bridge         Address:70:72:cf:3b:ea:a4    Priority:32768
Root           Address:70:72:cf:3b:ea:a4    Priority:32769
               Port:, Cost:0, Rem Hops:0
```

```
Port           Role           State      Cost    Priority   Type
-------------- -------------- ---------- ------- ---------- ----------
1              Designated     Forwarding 0       128        point_to_point
2              Designated     Forwarding 0       128        point_to_point
```

```
====================================================
[End] Feature mstp
====================================================
```

```
====================================================
Show Tech commands executed successfully
====================================================
```

2.4.9. MSTP events description

Event	Example
MSTP Enabled	Spanning Tree Protocol enabled
MSTP Disabled	Spanning Tree Protocol disabled
config_parameter should be value	FWD_DELAY should be >= 10
BPDU has config_parameter from port value	BPDU has FWD_DELAY 0 from port 1
Root changed from old_priority:old_mac to new_priority:new_mac	Root changed from 8:23:0f:cf:ed:c3:51 to 8:48:0f:cf:af:d3:93
Port port disabled - BPDU received on protected port	Port 1 disabled - BPDU received on protected port
proto starved for pkt_type on port port from priority_mac	CIST starved for BPDU rx on port 1 from 8:48:0f:cf:af:d3:93
BPDU loss- port port moved to inconsistent state for proto	BPDU loss- port 1 moved to inconsistent state for CIST
Topology Change received on port port for proto from source: mac	Topology Change received on port 1 for CIST from source: 48:0f:cf:af:d3:93
proto - Topology Change generated on port port going in to state	CIST - Topology Change generated on port 1 going in to forwarding
BPDU received on admin edge port	BPDU received on admin edge port 1
Port blocked on CIST	Port 1 blocked on CIST
Port unblocked on CIST	Port 1 unblocked on CIST
Port blocked on MST_instance	Port 1 blocked on MST2
Port unblocked on MST_instance	Port 1 unblocked for MST2
proto Root Port changed from old_port to new_port	MSTP Root port changed from 2 to 1

2.5. LLDP feature

2.5.1. Overview

The Link Layer Discovery Protocol (LLDP) is an industry-standard, vendor-neutral method to allow networked devices to advertise capabilities, discover and identify other LLDP enabled devices, and gather information in a LAN. The following bullet list contains some of the information gathered by LLDP:

- System name and description
- Port name and description
- VLAN name and identifier
- IP network management address
- Device Capabilities (for example, switch, route r, or server)
- MAC address and physical layer information
- Power information

2.5.2. Prerequisites

All the DUT interfaces (at least the interfaces that are connected to other devices) must be administratively up.

2.5.3. Setting up the basic configuration

1. Configure the terminal to change the vtysh context to config context with the following commands:

   ```
   switch# configure terminal
   switch(config)#
   ```

2. Enable LLDP globally on the switch with the following command:

   ```
   switch(config)# lldp enable
   switch(config)#
   ```

   LLDP is enabled by default and the switch begins to transmit advertisements from those ports that are configured to send LLDP packets.

3. Enable LLDP on interface.

   By using the lldp transmit and lldp receive commands, LLDP can be enabled or disabled on individual interfaces or configured to only send or only receive LLDP packets. Consider interface 1 which is connected to neighbor device,

   ```
   switch(config)# interface 1
   switch(config-if)# lldp receive
   switch(config-if)#
   ```

```
switch(config-if)# lldp transmit
switch(config-if)#
```

2.5.4. Setting up optional configurations

1. Setting the LLDP Timer.

   The lldp timer command specifies the time in seconds between LLDP updates sent by the switch.

   ```
   switch(config)# lldp timer 120
   switch(config)#
   ```

   The no lldp timer commands reverts the LLDP timer to its default value of 30 seconds.

   ```
   switch(config)# no lldp timer
   switch(config)#
   ```

2. Setting the LLDP Hold Time.

   The lldp holdtime command sets the number of transmit cycles for which receiving device retains the information sent by the device. Each transmit cycle duration will be that specified by the transmit timer.

   ```
   switch(config)# lldp holdtime 5
   switch(config)#
   ```

   The no lldp holdtime commands reverts the LLDP timer to its default value of four seconds.

   ```
   switch(config)# no lldp holdtime
   switch(config)#
   ```

3. Set the LLDP reinitialization delay value.

   The lldp reinit command sets the amount of time in seconds to wait before performing LLDP initialization on any interface.

   ```
   switch(config)# lldp reinit 5
   switch(config)#
   ```

   The no lldp reinit command reverts the reinitialization delay value to its default of two seconds.

   ```
   switch(config)# no lldp reinit
   switch(config)#
   ```

4. Set the IP address to be used in the Management Address TLV.

   The lldp management-address command specifies the IP address used in the management address LLDP type-length-value (TLV) triplets. If this command is not configured, the IP address assigned to the management interface is used in the management address LLDP type-length-value (TLV) triplets.

   ```
   switch(config)# lldp management-address 16.93.49.1
   switch(config)#
   ```

5. Select LLDP TLV.

   The lldp select-tlv command configures the type, length, and value (TLV) to send in LLDP packets. The no lldp select-tlv command removes the TLV configuration.

   ```
   switch(config)# lldp select-tlv system-capabilities
   switch(config)#
   switch(config)# lldp select-tlv port-description
   switch(config)#
   ```

6. Clearing the LLDP Counters.

   The lldp clear counters command resets the LLDP traffic counters to zero.

   ```
   switch(config)# lldp clear counters
   switch(config)#
   ```

7. Clearing the LLDP neighbor information.

   The lldp clear neighbors command clears neighbor information.

   ```
   switch(config)# lldp clear neighbors
   switch(config)#
   ```

2.5.5. Viewing LLDP global information

The **show lldp configuration** command displays LLDP configuration information configured above.

```
switch# show lldp configuration
LLDP Global Configuration:
LLDP Enabled :Yes
LLDP Transmit Interval :120
LLDP Hold time Multiplier :4
TLVs advertised:
Management Address
Port description
Port VLAN-ID
Port Protocol VLAN-ID
Port VLAN Name
Port Protocol-ID
System capabilities
System description
System name
LLDP Port Configuration:
Port  Transmission-enabled     Receive-enabled
1            Yes                    Yes
10           Yes                    Yes
```

2.5.6. Viewing LLDP neighbors

The **show lldp neighbor-info** command displays information about LLDP neighbors.

```
switch# show lldp neighbor-info
Total neighbor entries : 0
Total neighbor entries deleted : 0
Total neighbor entries dropped : 0
Total neighbor entries aged-out : 0
 Local Port     Neighbor Chassis-ID      Neighbor Port-ID         TTL
1
2
3
.......................
.......................
```

The show lldp neighbor-info <interface> command shows LLDP neighbors for a particular interface.

```
switch# show lldp neighbor-info 1
Port                           : 1
Neighbor entries               : 0
Neighbor entries deleted       : 0
Neighbor entries dropped       : 0
Neighbor entries age-out       : 0
Neighbor Chassis-Name          :
Neighbor Chassis-Description   :
Neighbor Chassis-ID            :
Chassis Capabilities Available :
Chassis Capabilities Enabled   :
Neighbor Port-ID               :
TTL                            :
switch#
```

2.5.7. Viewing LLDP statistics

The **show lldp statistics** command displays the LLDP traffic information for the switch.

```
switch# show lldp statistics
LLDP Global statistics:
Total Packets transmitted : 35
Total Packets received : 0
Total Packet received and discarded : 0
Total TLVs unrecognized : 0
LLDP Port Statistics:
Port-ID   Tx-Packets     Rx-packets     Rx-discarded        TLVs-Unknown
1              34            0                 0                    0
10              0            0                 0                    0
................
................
```

The show lldp statistics <interface> command shows LLDP traffic information for a particular interface.

```
switch# show lldp statistics 1
LLDP statistics:
```

```
Port Name: 1
Packets transmitted :36
Packets received :0
Packets received and discarded :0
Packets received and unrecognized :0
switch#
```

2.5.8. Viewing LLDP TLVs

The show lldp tlv command displays the LLDP TLVs to be sent and received.

```
switch# show lld tlv
TLVs advertised:
Management Address
Port description
Port VLAN-ID
Port Protocol VLAN-ID
Port VLAN Name
Port Protocol-ID
System capabilities
System description
System name
switch#
```

2.5.9. Viewing LLDP local device information

The show lldp local-device command displays the information advertised by the switch if the LLDP feature is enabled by the user. For example:

```
switch# show lldp local-device
Global Data
---------------
```

```
Chassis-id              : 48:0f:cf:af:50:c9
System Name            : switch
Systen Description     : OpenSwitch 0.1.0 (basil) Linux 3.9.11 #1 SMP Fri Sep 11 19:
Management Address     : 120.92.155.52
Capabilities Available : Bridge, Router
Capabilities Enabled   : Bridge, Router
TTL                    : 120
```

```
Port Based Data:
----------------
Port-ID           : 1
Port-Description  : "1"
Port VLAN Id      : 100
VLAN-Ids          : 100
VLAN Name         : VLAN100
```

2.5.10. Troubleshooting the configuration

Condition

- LLDP Neighbor information is not displayed even if the neighbor is present.
- System description is not displayed in neighbor information.

Cause

- Interface may be down.
- Neighbor may not support LLDP or the feature is not enabled.
- System description TLV may not be selected.

Remedy

- Make the interface administratively up by using *no shutdown* command. Refer to the physical interface command reference. The neighbor supports the LLDP feature and the enablement.
- Select system description TLV using *lldp select-tlv* command.

2.6. Unidirectional Link Detection feature

The UDLD supports two modes: normal and aggressive.

In normal mode, a port’s state is classified as undetermined if an anomaly exists. An anomaly might be the absence of its own information in received UDLD messages or the failure to receive UDLD messages. An undetermined state has no effect on the operation of the port. The port is not disabled and continues operating. When operating in UDLD normal mode, a port will be put into a disabled state (D-Disable) only in the following situations:

- The UDLD PDU received from a partner does not have its own details (echo).
- When there is a loopback, and information sent out on a port is received back exactly as it was sent.

When operating in UDLD aggressive mode, a port is put into a disabled state for the same reasons that it occurs in normal mode. Additionally, a port in UDLD aggressive mode can be disabled if the port does not receive any UDLD echo packets even after bidirectional connection was established. If a bidirectional link is established, and packets suddenly stop coming from partner device, the UDLD aggressive-mode port assumes that link has become unidirectional.

2.6.1. UDLD and LAG Interfaces

UDLD is supported on individual physical ports that are members of a LAG. If any of the aggregated links becomes unidirectional, UDLD detects it and disables the individual link, but not the entire LAG. This improves the fault tolerance of the LAG.

2.6.2. Configuring UDLD

A network administrator decides to use the UDLD feature while building a loop-free topology with the use of STP. The administrator configures the ports on both side of the link to use UDLD in aggressive mode to ensure that ports with unidirectional links will be shut down, and no loops will be introduced into topology. This example shows the steps to configure UDLD on Switch 1 only. The same configuration must be performed on all ports that form partner links with the ports on Switch 1.

2.7. Storm-Control feature

When Layer 2 frames are forwarded, broadcast, unknown unicast, and multicast frames are flooded to all ports on the relevant virtual local area network (VLAN). The flooding occupies bandwidth, and loads all nodes connected on all ports. Storm control limits the amount of broadcast, unknown unicast, and multicast frames accepted and forwarded by the switch.

Per-port and per-storm control type (broadcast, multicast, or unicast), the storm control feature can be configured to automatically shut down a port when a storm condition is detected on the port; or to send a trap to the system log. When configured to shut down, the port is put into a diag-disabled state. The user must manually re-enable the interface for it to be operational. When configured to send a trap, the trap is sent once in every 30 seconds. When neither action is configured, the switch rate-limits the traffic when storm conditions occur.

Chapter 3. Layer 3 features

3.1. L3 Interfaces

3.1.1. Overview

This document provides a step-by-step reference for configuration of basic layer3 functionality and features.

3.1.2. Layer3 interfaces

OpenSwitch supports configuring IPv4 and IPv6 addresses to layer3 interfaces. Every layer3 interfaces is associated with one VRF. In the first version, only one VRF is supported and hence the association with the VRF is not necessary.

To configure an interface:

```
switch# configure terminal
switch(config)# interface 1
switch(config-if)# no shutdown
switch(config-if)# ip address 192.168.1.1/24
switch(config-if)# ipv6 address 2000::1/120
switch(config-if)# end
switch#
```

3.1.3. VLAN interfaces

To achieve interVLAN routing, VLAN interfaces are created. VLAN interfaces are configured similar to physical interfaces. In addition to configuring IPv4 or IPv6 addresses, these interfaces are associated with a VLAN. The VLAN ID is part of the name of the VLAN interface being configured.

To configure a VLAN interface for VLAN ID 100:

```
switch# configure terminal
switch(config)# interface vlan100
switch(config-if)# no shutdown
switch(config-if)# ip address 192.168.1.1/24
switch(config-if)# ipv6 address 2000::1/120
switch(config-if)# end
switch#
```

To view the list of interfaces configured:

```
switch# show interface
Interface 1 is up
Admin state is up
Hardware: Ethernet, MAC Address: 70:72:cf:77:06:df
IPv4 address 192.168.1.1/24
MTU 0
Full-duplex
Speed 1000 Mb/s
Auto-Negotiation is turned on
```

```
Input flow-control is off, output flow-control is off
RX
           0 input packets              0 bytes
           0 input error                0 dropped
           0 CRC/FCS
TX
           0 output packets             0 bytes
           0 input error                0 dropped
           0 collision
switch# show ip interface
Interface 1 is up
Admin state is up
Hardware: Ethernet, MAC Address: 70:72:cf:77:06:df
IPv4 address: 192.168.1.1/24
MTU 0
RX
         ucast: 10 packets, 750 bytes
         mcast: 0 packets, 0 bytes
TX
         ucast: 10 packets, 750 bytes
         mcast: 0 packets, 0 bytes
switch# show interface brief
--------------------------------------------------------------------------------
Ethernet      VLAN    Type Mode   Status  Reason                   Speed     Port
Interface                                                          (Mb/s)    Ch#
--------------------------------------------------------------------------------
1            --      eth  --     up                               1000     --
```

3.1.4. LAG interfaces

LAG interfaces are created to achieve LAG routing. LAG L3 interfaces are configured similar to LAG L2 interfaces, in addition to configuring IPv4 or IPv6 addresses.

Execute the following commands to configure a LAG routing interface:

```
switch# configure terminal
switch(config)# interface lag 1
switch(config-if)# routing
switch(config-if)# ip address 192.168.1.1/24
switch(config-if)# ipv6 address 2000::1/120
switch(config-if)# end
switch#
```

Execute the following command to view the configured interface:

```
switch# show interface lag1
Aggregate-name lag1
Aggregated-interfaces :
Aggregation-key : 1
IPv4 address 192.168.1.1/24
IPv6 address 2000::1/120
Speed 0 Mb/s
```

```
RX
           0 input packets              0 bytes
           0 input error                0 droppd
           0 CRC/FCS
TX
           0 output packets             0 bytes
           0 input error                0 dropped
           0 collision
```

3.1.5. Static routes

OpenSwitch supports the configuration of IPv4 and IPv6 static routes.

To add static routes:

```
switch# configure terminal
switch(config)# ip route 192.168.2.0/24 192.168.1.2
switch(config)# ipv6 route 2002::0/120 2000::1
```

To see the active routes in the system, including static routes:

```
switch# show ip route
Displaying ipv4 routes selected for forwarding
'[x/y]' denotes [distance/metric]
10.0.30.0/24,  1 unicast next-hops
       via  3,  [0/0],  connected
10.0.20.0/24,  1 unicast next-hops
       via  2,  [0/0],  connected
10.0.10.0/24,  1 unicast next-hops
       via  1,  [0/0],  connected
10.0.40.0/24,  1 unicast next-hops
       via  4,  [0/0],  connected
10.0.70.0/24,  2 unicast next-hops
       via  10.0.40.2,  [1/0],  static
       via  10.0.30.2,  [1/0],  static
switch#
```

To see all the routes in the system:

```
switch# show rib
Displaying ipv4 rib entries
'*' denotes selected
'[x/y]' denotes [distance/metric]
*10.0.30.0/24,  1 unicast next-hops
       *via  3,  [0/0],  connected
*10.0.20.0/24,  1 unicast next-hops
       *via  2,  [0/0],  connected
*10.0.10.0/24,  1 unicast next-hops
       *via  1,  [0/0],  connected
*10.0.40.0/24,  1 unicast next-hops
       *via  4,  [0/0],  connected
*10.0.70.0/24,  2 unicast next-hops
```

```
       *via  10.0.40.2,  [1/0],  static
       *via  10.0.30.2,  [1/0],  static
No ipv6 rib entries
switch#
```

3.1.6. ECMP

ECMP capability in OpenSwitch is currently available for IPv4 and IPv6 routes. ECMP is enabled by default.

By default, ECMP uses four tuple in the hash calculation:

- SrcIP
- DstIP
- SrcPort
- DstPort

Each of these fields can be explicitly disabled from the hash calculations.

For example, to disable the dst-ip field from the ECMP hash calculation use the following commands:

```
switch# configure terminal
switch(config)# ip ecmp load-balance dst-ip disable
```

To add back dst-ip in the hash calculation:

```
switch# configure terminal
switch(config)# no ip ecmp load-balance dst-ip disable
```

To see all the tuples that can be added/removed from the ECMP hash:

```
switch# configure terminal
switch(config)# ip ecmp load-balance ?
 dst-ip    Load balancing by destination IP
 dst-port  Load balancing by destination port
 src-ip    Load balancing by source IP
 src-port  Load balancing by source port
```

If the platform supports it, OpenSwitch will enable "resilient ECMP" by default to preserve in-flight traffic flows when ECMP group membership changes.

To disable resilient ECMP, use the following commands:

```
switch# configure terminal
switch(config)# ip ecmp load-balance resilient disable
```

To re-enable resilient ECMP:

```
switch# configure terminal
switch(config)# no ip ecmp load-balance resilient disable
```

To see the complete ECMP configuration:

```
switch# show ip ecmp
ECMP Configuration
---------------------
ECMP Status        : Enabled
Resilient Hashing  : Enabled
ECMP Load Balancing by
-----------------------
Source IP          : Enabled
Destination IP     : Enabled
Source Port        : Enabled
Destination Port   : Enabled
```

3.1.7. Internal VLAN management

Every layer3 interface is associated with a unique VLAN ID. By default, OpenSwitch uses VLAN IDs from the range 1024-4094 for this purpose. However, this range is configurable. The order in which the VLAN IDs are used in this range is also be specified using "ascending" or "descending" in the CLI.

To configure the VLAN range for internal use:

```
switch# configure terminal
switch(config)# vlan internal range 400 500 ascending
```

To show the configured VLAN range

```
switch# sh vlan internal
Internal VLAN range  : 400-500
Internal VLAN policy : ascending
------------------------
Assigned Interfaces:
       VLAN            Interface
       ----            ---------
       401             2
       400             1
switch#
```

3.2. Loopback Interface feature

3.2.1. Overview

A loopback interface is a virtual interface that supports IPv4 and IPv6 address configurations and remains running until you disable it. Unlike subinterfaces, loopback interfaces are independent of the state of any physical interface. For example, Router IDs are for routing protocols like OSPF.

The loopback interface can be considered stable because once enabled, it will remain running until you shut it down. This makes loopback interfaces ideal for assigning Layer 3 addresses such as IP addresses when you want a single address as a reference that is independent of the status of any physical interfaces in the networking device.

The maximum limit of loopback interfaces is 1024.

3.2.2. Setting up the basic configuration

1. Create a loopback interface.
2. Set up the IPv4 or IPv6 addresses.
3. Enable the loopback interface.

3.2.3. Verifying the configuration

Display the configured loopback interfaces.

3.2.4. Troubleshooting the configuration

Condition

Unable to ping the loopback interface from an external entity.

Cause

An overlapping IP address is set on the loopback interface.

Remedy

Check for solutions in the /var/log/messages file.

3.2.5. Loopback event logs

All the events related to loopback configuration are logged in event log.

Following are the logged events:

- Create loopback interface.
- Configure loopback interface with IPv4 address.

- Configure loopback interface with IPv6 address.
- Remove IPv4 address from loopback inetrface.
- Remove IPv6 address from loopback interface.
- Delete loopback interface.

3.2.6. Loopback diagnostic dump

Number of loopback interfaces created can be dumped using diagnostic dump.

3.2.7. Loopback interface show tech

Configurations done for loopback interfaces can be seen from show tech.

3.3. ARP feature

You can create static Address Resolution Protocol (ARP) entries and manage many settings for the dynamic ARP table, such as age time for entries, retries, and cache size.

Proxy ARP is a technique by which a device on a given network answers the ARP queries for a network address that is not on that network. The ARP proxy is aware of the location of the traffic’s destination, and offers its own MAC address as the final destination.

3.4. L3 Subinterfaces feature

3.4.1. Overview

L3 Subinterfaces are used to support router-on-a-stick configurations. Using router-on-a-stick configurations, you can separate traffic on a L3 physical interface based on VLAN and also apply policies on the subinterfaces.

An example use of L3 subinterfaces in a data center deployment is shown in this diagram. An L3 interface of a TOR switch is connected to the trunk port of a switch. All the outgoing traffic from the L3 interface of the TOR switch is tagged with a VLAN ID. This enables the switch to forward the traffic on different VLANs. This is configured by creating L3 subinterfaces on the TOR switch and configuring the routing tables to forward the outgoing traffic on one of these subinterfaces while applying a different VLAN tag on each subinterface.

```
+-------------------------+                   +---------------------------+
|                         |                   |                           |
|                         |                   |                           |
|                         |                   |                           |
|  OpenSwitch            +-+ VLAN1(Subintf1) +-+         L2 Switch        |
|                 L3 I/F |||<--------------->|||Trunk                     |
|                        |||<--------------->|||Port                      |
|                        +-+ VLAN2(Subintf2) +-+                          |
|                         |                   |                           |
|                         |                   |                           |
|                         |                   |                           |
+-------------------------+                   +---------------------------+
```

3.4.2. Subinterface restrictions

A subinterface cannot be assigned an IP address already used by any interfaces on the switch.

3.4.3. Setting up the basic configuration

1. Configure an interface as L3.
2. Run the no shut command on the parent interface.
3. Create a subinterface on the L3 interface.
4. Assign dot1q encapsulation.
5. Assign an IP address.
6. Run the no shut command.
7. Enable the subinterface.

3.4.4. Verifying the configuration

Display the configured subinterfaces.

3.4.5. Troubleshooting the configuration

Condition

Unable to create a subinterface.

Cause

The interface may be configured as an L2.

Remedy

Configure the interface as an L3 using the routing command.

3.4.6. L3 subinterface event logs

All the events related to subinterface configuration are logged in event log.

Following are the logged events:

- Create subinterface.
- Configure subinterface with IPv4 address.
- Configure subinterface with IPv6 address.
- Configure subinterface with encapsulation dot 1Q vlan ID.
- Configure subinterface with admin up.
- Configure subinterface with admin down.
- Remove IPv4 address.
- Remove IPv6 address.
- Remove encapsulation dot 1Q vlan ID.
- Delete subinterface.

3.4.7. L3 subinterface diagnostic dump

Number of subinterfaces created can be dumped using diagnostic dump.

3.5. UDP Broadcast Forwarder

3.5.1. Overview

The routers by default do not forward broadcast packets. This is to avoid packet flooding on the network. However, there are situations where it is desirable to forward certain broadcast packets. The UDP (User Datagram Protocol) broadcast forwarder takes up the client’s UDP broadcast packet and forwards it to the configured server(s) in a different subnet. By default, a router’s UDP broadcast forwarding is disabled. A client’s UDP broadcast requests cannot reach a target server on a different subnet unless explicitly configured on the router to forward client UDP broadcasts to that server. UDP forward-protocol addresses can be configured on an interface regardless of whether UDP broadcast forwarding is globally enabled on the device. However, the feature does not operate unless globally enabled. A UDP forwarding entry includes the application UDP port number, and either an IP unicast address or an IP subnet broadcast address on which the server operates. Thus, an incoming UDP packet carrying the configured port number will be:

1. Forwarded to a specific host if a unicast server address is configured for that port number.

2. Broadcast on the appropriate destination subnet if a subnet address is configured for that port number.

UDP broadcast forwarder allows multiple unicast server IPs to be configured for a single UDP port. UDP broadcast forwarding is supported only for IPv4 addresses. UDP forward-protocol configuration is allowed on the interface with routing disabled, but UDP broadcast forwarding will not take effect on that interface until routing is enabled.

3.5.2. Configure the UDP broadcast forwarder

Syntax [no] ip udp-bcast-forward Enable/disable UDP broadcast forwarding. By default, it is disabled.

Syntax [no] ip forward-protocol udp <IPv4-address> <port-number **|** protocol-name> Configure UDP broadcast server(s) on the interface for a particular udp port.

UDP forward-protocol configuration is supported on data-plane interfaces.

Explanation of parameters:

- IPv4-address - The IPv4 address of the protocol server. This can be either be a unicast address of a destination server on another subnet or a broadcast address of the subnet on which a destination server operates.

- port-number - Any UDP port number corresponding to a UDP application supported on a device.

- protocol-name - Allows the use of common names for certain well-known UDP port numbers.

Supported UDP protocols:

- dns: Domain Name Service (53)

- ntp: Network Time Protocol (123)

- netbios-ns: NetBIOS Name Service (137)

- netbios-dgm: NetBIOS Datagram Service (138)
- radius: Remote Authentication Dial-In User Service (1812)
- radius-old: Remote Authentication Dial-In User Service (1645)
- rip: Routing Information Protocol (520)
- snmp: Simple Network Management Protocol (161)
- snmp-trap: Simple Network Management Protocol (162)
- tftp: Trivial File Transfer Protocol (69)
- timep: Time Protocol (37)

Syntax show ip forward-protocol [interface <WORD>] Display the server addresses where broadcast requests received by the device are to be forwarded based on configured port.

Explanation of parameters

- interface - The interface on which server addresses are configured.

3.5.3. How to use the UDP broadcast forwarder

Example

```
switch(config)#ip udp-bcast-forward
switch#show ip forward-protocol
UDP Broadcast Forwarder : enabled
switch#show running-config
Current configuration:
!
!
!
ip udp-bcast-forward
switch(config)#interface 1
switch(config-if)#ip forward-protocol udp 1.1.1.1 53
switch#show ip forward-protocol
UDP Broadcast Forwarder : enabled
Interface: 1
 IP Forward Address    UDP Port
 -----------------------------
 1.1.1.1                53
switch#show ip forward-protocol interface 1
UDP Broadcast Forwarder : enabled
Interface: 1
IP Forward Address    UDP Port
 -------------------------------
 1.1.1.1                53
switch#show running-config
Current configuration:
```

```
!
!
!
interface 1
   ip forward-protocol udp 1.1.1.1 53
ip udp-bcast-forward
```

```
switch(config)#interface 1
switch(config-if)#ip forward-protocol udp 8.1.1.1 161
switch(config-if)#ip forward-protocol udp 4.4.4.4 137
switch(config)#interface 2
switch(config-if)#ip forward-protocol udp 3.3.3.3 137
switch#show ip forward-protocol
UDP Broadcast Forwarder : enabled
Interface: 1
 IP Forward Address    UDP Port
 -----------------------------
 4.4.4.4                137
 1.1.1.1                53
 8.1.1.1                161
Interface: 2
 IP Forward Address    UDP Port
 -----------------------------
 3.3.3.3                137
switch#show ip forward-protocol interface 1
UDP Broadcast Forwarder : enabled
Interface: 1
 IP Forward Address    UDP Port
 -------------------------------
 4.4.4.4                137
 8.1.1.1                161
 1.1.1.1                53
switch#show running-config
Current configuration:
!
!
!
!
interface 1
   ip forward-protocol udp 1.1.1.1 53
   ip forward-protocol udp 8.1.1.1 161
   ip forward-protocol udp 4.4.4.4 137
interface 2
   ip forward-protocol udp 3.3.3.3 137
ip udp-bcast-forward
```

```
switch(config)#no ip udp-bcast-forward
switch#show ip forward-protocol
UDP Broadcast Forwarder : disabled
Interface: 1
 IP Forward Address    UDP Port
 -----------------------------
 4.4.4.4                137
 1.1.1.1                53
 8.1.1.1                161
Interface: 2
 IP Forward Address    UDP Port
 -----------------------------
 3.3.3.3                137
```

```
switch#show ip forward-protocol interface 1
UDP Broadcast Forwarder : disabled
Interface: 1
 IP Forward Address    UDP Port
 -------------------------------
 4.4.4.4                137
 1.1.1.1                53
 8.1.1.1                161
switch#show running-config
Current configuration:
!
!
!
interface 1
   ip forward-protocol udp 1.1.1.1 53
   ip forward-protocol udp 8.1.1.1 161
   ip forward-protocol udp 4.4.4.4 137
interface 2
   ip forward-protocol udp 3.3.3.3 137
```

```
switch(config)#interface 1
switch(config-if)#no ip forward-protocol udp 1.1.1.1 53
switch#show ip forward-protocol
UDP Broadcast Forwarder : disabled
Interface: 1
 IP Forward Address    UDP Port
 -----------------------------
 4.4.4.4                137
 8.1.1.1                161
Interface: 2
 IP Forward Address    UDP Port
 -----------------------------
 3.3.3.3                161
switch#show ip forward-protocol interface 1
UDP Broadcast Forwarder : disabled
Interface: 1
IP Forward Address    UDP Port
 -------------------------------
 4.4.4.4                137
 8.1.1.1                161
switch#show running-config
Current configuration:
!
!
```

```
!
interface 1
   ip forward-protocol udp 8.1.1.1 161
   ip forward-protocol udp 4.4.4.4 137
interface 2
   ip forward-protocol udp 3.3.3.3 137
```

3.6. eBGP feature

3.6.1. Overview

The Border Gateway Protocol (BGP) is the most commonly used as an inter-AS (autonomous system) routing protocol. The latest BGP version is 4. BGP-4 supports Classless Inter-Domain Routing (CIDR). The BGP advertises the routes based on destinations as an IP prefix and not the network "class" within BGP.

BGP is a path-vector protocol that provides routing information across various BGP routers to be exchanged using destination-based forwarding. For example, the router sends packets based merely on the destination IP address carried in the IP header of the packet. In most cases, there are multiple routes to the same destination, BGP decides which route to choose using the path attributes, such as Shortest AS_Path, Multi_Exit_Disc (Multi-exit discriminator, or MED), Origin, Next_hop, Local_pref, and so on.

eBGP provides routing for routers or switches or both, that can be deployed in ISP, Enterprise, and data center environments.

3.6.2. Setting up the basic configuration

The following is the minimum configuration needed to set up the eBGP router. The AS number is unique to the autonomous system, and is used to distinguish between internal, external, or both eBGP connections. Enter the following commands in the order shown:

1. The **router bgp <asn>** command enables the eBGP router for the AS number. The AS number ranges from 1 to 65535.

2. The **bgp router-id A.B.C.D** command sets the eBGP router-id.

The eBGP router can be disabled with the no bgp router-id A.B.C.D or no bgp router-id commands. The commands default the router-id to 0.0.0.0.

The no router bgp <asn> command disables the eBGP process with the given AS number.

3.6.3. Setting up the optional configuration

To set up the optional configuration:

- Enter the router bgp command with the required AS number. For example: router bgp <asn> The router bgp <asn> command enables the eBGP router for the AS number. The AS numbers range from 1 to 65535.

- Set the eBGP router id with the following command: bgp router-id <A.B.C.D> The eBGP router can be disabled with the no bgp router-id A.B.C.D or no bgp router-id commands. The commands default the router-id to 0.0.0.0.

- Set up the maximum-paths with the following commands: router bgp <asn>, maximum-paths <paths> The maximum-paths command limits the maximum number of paths for eBGP. If global ECMP is enabled, and eBGP maximum-paths is set greater than the global maximum-paths, then the global setting overrides eBGP maximum-paths. If global ECMP is disabled, then only single best path gets selected. eBGP multiple-paths carries risks of routing oscillations if MED, IGP costs, and the eBGP and IGP topologies are not cautiously considered. Since community and extended community are the aggregated attributes of the multi-path routes within AS path, eBGP multi-pathing results in the propagated route having the attributes of the "best route" of the multi-path. The no maximum-paths defaults the number of maximum paths to 1.

- Set up timers with the following commands: router bgp <asn>, timers bgp <keepalive> <holdtimer> The timers bgp <keepalive> <holdtimer> command sets the keepalive interval and hold time for the eBGP router. Timers can be set to default with the no timers bgp <keepalive> <holdtimer> command. The default keepalive interval is 180 seconds, and the default hold time is 60 seconds.

- To add a static network to the eBGP routing table, enter the following commands: router bgp <asn>, network <A.B.C.D/M> It announces the specified network to all peers in the AS. The no network <A.B.C.D/M> command removes the announced network for this eBGP router.

- Use the following commands to advertise the IPv6 prefix network: router bgp <asn>, network <X:X::X:X/M> Use this command in router configuration mode to advertise the specified prefix into the IPv6 eBGP database. Use the no network <X:X::X:X/M> command to stop advertising the specified prefix into the IPv6 eBGP database.

- Use the following commands to enable fast external failover for eBGP directly connected peering sessions: router bgp <asn>, bgp fast-external-failover Use this command in router configuration mode to terminate external eBGP sessions of any directly adjacent peer if the link used to reach the peer goes down, without waiting for the hold-down timer to expire. Use the no bgp fast-external-failover command to disable the eBGP fast external failover.

- Use the following commands to enable logging of eBGP neighbor status changes: router bgp <asn>, bgp log-neighbor-changes Use this command in router configuration mode to log changes in the status of eBGP neighbors (up, down, reset). Use the no bgp log-neighbor-changes command to disallow log changes in the status of eBGP neighbors.

- Enter the following neighbor configuration commands in the order shown:

- Use the following commands to define a new peer with remote-as, where the asn and peer parameters are IPv4 addresses: router bgp <asn>, neighbor <peer> remote-as <asn> If remote-as is not configured, bgpd displays the error can’t find neighbor peer. The no neighbor <peer> command deletes the peer.

- Set up the peer description with the following commands: router bgp <asn>, neighbor <peer> description <some_description> The no neighbor <peer> description command deletes the neighbor description.

- Enable MD5 authentication on a TCP connection between eBGP peers with the following command: router bgp <asn>, neighbor <peer> password <some_password> The no neighbor <peer> password <some_password> command disables MD5 authentication on a TCP connection between eBGP peers.

- Set up the keepalive interval or hold time for a peer with the following commands: router bgp <asn>, neighbor peer timers <keepalive> <holdtimer> The no neighbor peer timers <keepalive> <holdtimer> command clears the keepalive interval and hold time for that peer.

- Specify the number of times eBGP allows an instance of AS to be in the AS path using the following commands: router bgp <asn>, neighbor peer allowas-in <ASN_instances_allowed> The ASN_instances_allowed range is from 1 to 10. The no neighbor peer allowas-in <ASN_instances_allowed> command prevents the AS number from being added to the AS path by setting the ASN_instances_allowed parameter to 0.

- Remove private AS numbers from the AS path in outbound routing updates with the following commands: router bgp <asn>, neighbor peer remove-private-AS The no neighbor peer remove-private-AS command allows private AS numbers from the AS path in outbound routing updates.

- Enable software-based reconfiguration to generate inbound updates from a neighbor without clearing the eBGP session with the following commands: router bgp <asn>, neighbor peer soft-reconfiguration inbound The no neighbor peer soft-reconfiguration inbound command disables the software-based reconfiguration.

- Set the advertisement interval for route updates for a specified neighbor or peer with an IPv4 or IPv6 address with the following commands: router bgp <asn>, neighbor peer advertisement-interval <interval> The time interval for sending eBGP routing updates is in the range of 0 to 600 seconds. The default value is 30 seconds. The no neighbor peer advertisement-interval <interval> command unsets the advertisement interval for route updates for the specified neighbor or peer with an IPv4 or IPv6 address.

- Configure a filter list on a neighbor to filter incoming and outgoing routes using the following commands: router bgp <asn>, neighbor (A.B.C.D|X:X::X:X|WORD) filter-list WORD (in|out) The no neighbor (A.B.C.D|X:X::X:X|WORD) filter-list WORD (in|out) command uninstalls the filter list.

- Apply a prefix list on a neighbor to filter updates to and from the neighbor using the following command: router bgp <asn>, neighbor (A.B.C.D|X:X::X:X|WORD) prefix-list WORD (in|out) The no neighbor (A.B.C.D|X:X::X:X|WORD) prefix-list WORD (in|out) command uninstalls the applied prefix list.

- Allow inbound soft reconfiguration for a neighbor using the following commands: router bgp <asn>, neighbor (A.B.C.D|X:X::X:X|WORD) soft-reconfiguration inbound The no neighbor (A.B.C.D|X:X::X:X|WORD) soft-reconfiguration inbound command disables the inbound soft reconfiguration.

- Specify the maximum number of hops to the eBGP peer using the following commands: router bgp <asn>, neighbor (A.B.C.D|X:X::X:X|WORD) ttl-security hops <1-254> The no neighbor (A.B.C.D|X:X::X:X|WORD) ttl-security hops <1-254> command disables the maximum number of hops specification.

- The peer-group is a collection of peers that share the same outbound policy. Neighbors belonging to the same peer-group might have different inbound policies. All peer commands are applicable to the peer-group as well. Following are the peer-group configuration commands. Enter the following BGP peer-group commands in the order shown:

- Define a new peer-group with the < word > variable as the name of the peer-group using the following commands: router bgp <asn>, neighbor <word> peer-group The neighbor <word> peer-group command deletes the peer-group.

- Bind a specific peer to the provided peer-group with the following commands: router bgp <asn>, neighbor peer peer-group word

- Following are the route map configuration commands:
- Configure peer filtering of a given sequence number for a route map with the following command: route-map <word> (deny|permit) <sequence_num> The word variable is the name of the route map, and the sequence_num variable is an integer in the range from 1 to 65535. The no route-map <word> (deny|permit) <sequence_num> command deletes the route map. All route map commands should be executed under the route-map <word> (deny|permit) <sequence_num> context.
- Assign a route map to the peer with the given direction using the following commands: route-map <word> (deny|permit) <sequence_num>, neighbor <peer> route-map <word> in|out The no neighbor <peer> route-map <word> in|out command removes the route map from the peer.
- Add a route map description with the following command: route-map <word> (deny|permit) <sequence_num>, description <some_route_map_description>
- Assign or change the community for a route map with the following command: route-map <word> (deny|permit) <sequence_num>, set community .AA:NN additive The no set community .AA:NN additive command removes the community.
- Set or change the metric for a route map with the following command: route-map <word> (deny|permit) <sequence_num>, set metric <0-4294967295> This command can be used to overwrite a previously set metric. The no set metric <0-4294967295> command resets the metric to 0.
- Set the metric value for the route, which is used with eBGP route advertisement, using the following commands: route-map <word> (deny|permit) <sequence_num>, set metric <value> The value attribute is in the range 0 - 4294967295. The no set metric <value> command resets the metric for the route to 0.
- To match an IP address prefix list, use the following commands: route-map <word> (deny|permit) <sequence_num>, match ip address prefix-list WORD Use this command in route map configuration mode to distribute any routes that have a destination network address that is permitted by the prefix list. The no match ip address prefix-list WORD command removes the match ip address prefix-list entry.
- Configure the IP prefix list for a given route map with the following command: ip prefix-list WORD seq <1-4294967295> (deny|permit) (A.B.C.D/M|any) The no ip prefix-list <word>seq <1-4294967295> (deny|permit) (A.B.C.D/M|any) command deletes the IP prefix list for the given route map.
- Configure the IP prefix list for a given route map with a prefix length specification using one of the following commands: ip prefix-list WORD seq <1-4294967295>deny|permit <A.B.C.D/M>ge <length> ip prefix-list WORD seq <1-4294967295>deny|permit <A.B.C.D/M>ge <length>le <length> ip prefix-list WORD seq <1-4294967295>deny|permit <A.B.C.D/M>le <length> The no ip prefix-list WORD command deletes the IP prefix list for the given route map.
- Configure the IPv6 prefix list for a given route map with the following command: ipv6 prefix-list WORD seq <1-4294967295>deny|permit <X:X::X:X/M|any> The no ipv6 prefix-list WORD command deletes the IPv6 prefix list for the given route map.
- Configure the IPv6 prefix list for a given route map with a prefix list description using the following command: ipv6 prefix-list WORD description .LINE The no ipv6 prefix-list WORD command deletes the IPv6 prefix list for the given route map.

- Configure the IPv6 prefix list for a given route map with prefix length specification using one of the following commands: ipv6 prefix-list WORD seq <1-4294967295>deny|permit <X:X::X:X/M>ge <length> ipv6 prefix-list WORD seq <1-4294967295>deny|permit <X:X::X:X/M>ge <length>le <length> ipv6 prefix-list WORD seq <1-4294967295>deny|permit <X:X::X:X/M>le <length> The no ipv6 prefix-list WORD command deletes the IPv6 prefix list for the given route map.

- Facilitate the configuration of access lists based on autonomous system paths that control routing updates based on eBGP autonomous paths information. Access lists are filters that restrict the routing information a router learns or advertises to and from a neighbor. Multiple eBGP peers or route maps can reference a single access list. These access lists can be applied to both inbound route updates and outbound route updates. Each route update is passed through the access-list. eBGP applies each rule in the access list in the order it appears in the list. When a route matches any rule, the decision to permit the route through the filter or deny is made, and no further rules are processed. Configure AS_PATH access lists using the following: ip as-path access-list WORD <deny | permit>.LINE LINE is a pattern used to match against an input string. In eBGP, a regular expression can be built to match information about an autonomous system path. The no ip as-path access-list WORD <deny|permit>.LINE command disables the access list configuration for AS_PATH.

- The community is compiled into a community structure. Multiple community lists can be defined under the same name. If that is the case, match happens in user-defined order. Once the community list matches to the communities attribute in eBGP updates, the system returns either a permit or a deny response based on the community list definition. When there is no matched entry, deny is returned. When the community is empty, the system matches to any routes. Define a new community list as shown: ip community-list WORD <deny|permit>.LINE The .LINE parameter is a string expression of communities attribute -→[1-2]00 , ^0:._, . The WORD parameter is a string. The no ip community-list expanded WORD <deny|permit>.LINE command disables community list for configured communities.

- Configure the extended community list with the following command: ip extcommunity-list WORD <deny|permit>.LINE The no ip extcommunity-list WORD command deletes the extended community list.

- Configure a prefix-list for a match on a given IP address with the following command: match ip address prefix-list <word>. The no match ip address prefix-list <word> command removes the rule for match on the IP address from the prefix list.

- The command neighbor <ipv4_address | ipv6_address | peer_group_name>ebgp-multihop attempts to connect to the external Autonomous System routers which are not directly connected. It takes either an IPv4 or IPv6 address or the peer-group name to establish a connection. To remove the ebgp-multihop configuration, use the no neighbor <ipv4_address | ipv6_address | peer_group_name>ebgp-multihop command.

3.6.4. Verifying the configuration

Use the **show running-config** command to verify the configuration. All active configurations are displayed with the show running-config command. See the sample output below:

```
 s1# show running-config
>Current configuration:
 !
```

```
ip prefix-list BGP1 seq 5 permit 11.0.0.0/8
ip prefix-list BGP1 seq 6 deny 12.0.0.0/8
!
route-map BGP2 permit 5
      description tsting route map description
      match ip address prefix-list bgp1
      set community 123:345 additive
      set metric 1000
!
router bgp 6001
      bgp router-id 9.0.0.1
      network 11.0.0.0/8
      maximum-paths 5
      timers bgp 3 10
      neighbor openswitch peer-group
      neighbor 9.0.0.2 remote-as 2
      neighbor 9.0.0.2 description abcd
      neighbor 9.0.0.2 password abcdef
      neighbor 9.0.0.2 timers 3 10
      neighbor 9.0.0.2 route-map BGP2 in
      neighbor 9.0.0.2 route-map BGP2 out
      neighbor 9.0.0.2 allowas-in 7
      neighbor 9.0.0.2 remove-private-AS
      neighbor 9.0.0.2 soft-reconfiguration inbound
      neighbor 9.0.0.2 peer-group openswitch
!
```

3.6.5. Troubleshooting the configuration

The following commands verify eBGP route related information:

- The show ip bgp command verifies that all the routes are advertised from the peers.

```
 s1# show ip bgp
 Status codes: s suppressed, d damped, h history, * valid,>best, = multipath,
        i internal, S Stale, R Removed
 Origin codes: i - IGP, e - EGP, ? - incomplete
 Local router-id 9.0.0.2
 Network      Next Hop     Metric LocPrf Weight Path
>11.0.0.0/8       9.0.0.1           0      0      0 1 i
>12.0.0.0/8       0.0.0.0           0      0  32768  i
 Total number of entries 2
```

- For more information about a specific peer, use the show command.

```
s1# show ip bgp 11.0.0.0/8
BGP routing table entry for 11.0.0.0/8
Paths: (1 available, best #1)
AS: 1
    9.0.0.1 from 9.0.0.1
Origin IGP, metric 0, localpref 0, weight 0, valid, external, best
Last update: Thu Sep 24 22:45:52 2015
```

- The show ip bgp summary command provides peer status and additional neighbor information such as eBGP packet statistics, total RIB entries, bgp router-id, and local AS number.

```
s1# show ip bgp summary
BGP router identifier 9.0.0.2, local AS number 2
RIB entri es 2
Peers 1
Neighbor      AS MsgRcvd MsgSent Up/Down  State
```

- The show ip bgp neighbors command provides detailed information about the neighbor such as neighbor state, description, tcp port number, password (if any), and statistics.

```
s1# show ip bgp neighbors
 name: 9.0.0.1, remote-as: 1
         state: Established
 description: abcd
 password: abcd
 tcp_port_number: 179
 statistics:
       bgp_peer_dropped_count: 1
       bgp_peer_dynamic_cap_in_count: 0
       bgp_peer_dynamic_cap_out_count: 0
       bgp_peer_established_count: 1
       bgp_peer_keepalive_in_count: 3
       bgp_peer_keepalive_out_count: 4
       bgp_peer_notify_in_count: 0
       bgp_peer_notify_out_count: 1
       bgp_peer_open_in_count: 1
       bgp_peer_open_out_count: 1
       bgp_peer_readtime: 25066
       bgp_peer_refresh_in_count: 0
       bgp_peer_refresh_out_count: 0
       bgp_peer_resettime: 25101
       bgp_peer_update_in_count: 2
       bgp_peer_update_out_count: 2
       bgp_peer_uptime: 25101
```

3.7. OSPFv2 feature

OSPFv2 (Open Shortest Path First version 2) is a routing protocol which is described in RFC2328 entitled OSPF Version 2. It is a Link State-based IGP (Interior Gateway Protocol) routing protocol. It is widely used with medium to large-sized enterprise networks.

3.8. Source Interface feature

3.8.1. Overview

The source interface selection is used to set the IP address of an interface, or IP address-defined interface as the source interface for the TFTP protocol or all the specified protocols.

Syntax	ip source-interface <protocol-ID | all> <interface <id>| address <ip-address>>
Syntax	[no] ip source-interface <protocol-ID | all> show ip source-interface [tftp]

Explanation of Parameters:

- protocol-ID - Specifies the different software applications like telnet, tftp, radius, sflow etc. we can specify different source ips for different apps by using protocol-ID
- all - Specifies same source IP for all applications. -address—Sets the IP address of an interface as the source IP. -interface—Sets an interface as the source interface.

As of now the CLI infra is ready, end to end functionality of source interface selection is not implemented.

Examples:

Configuring the source-interface

- Configuring a source-interface IP address to TFTP protocol

```
switch(config)# ip source-interface tftp address 1.1.1.1
```

- Configuring a source-interface IP address to all the specified protocols

```
switch(config)# ip source-interface all address 1.1.1.1
```

- Configuring a source-interface to TFTP protocol

```
switch(config)# ip source-interface tftp interface 1
```

- Configuring a source-interface to all the specified protocols

```
switch(config)# ip source-interface all interface 1
```

- Unconfigure the source-interface from the TFTP protocol.

```
switch(config)# no ip source-interface tftp
```

- Unconfigure the source-interface from all the specified protocols.

```
switch(config)# no ip source-interface all
```

3.8.2. Viewing source-interface information

- Verify that the source-interface is on the TFTP protocol.

```
switch# show ip source-interface tftp
Source-interface Configuration Information
Protocol        Source Interface
-------        ----------------
tftp            1.1.1.1
```

- Verify that source-interface to all the specified protocols

```
switch# show ip source-interface
Source-interface Configuration Information
Protocol        Source Interface
--------        ----------------
tftp            1.1.1.1
```

- Verify that the source-interface from the TFTP protocol.

```
switch# show ip source-interface tftp
Source-interface Configuration Information
Protocol        Source Interface
--------        ----------------
tftp
```

- Verify that unconfiguring the source-interface from all the specified protocols.

```
switch# show ip source-interface
Source-interface Configuration Information
Protocol        Source Interface
--------        ----------------
tftp
```

3.8.3. Viewing the snapshot of active configurations.

```
 switch# show ip source-interface
 Source-interface Configuration Information
 Protocol        Source Interface
 --------        ----------------
 tftp            1.1.1.1
 switch# show running-config interface
 Current configuration:
 !
 !
 !
 interface 1
    no shutdown
    ip address 1.1.1.1/24
source interface
    1.1.1.1
```

3.9. Virtual Router Redundancy Protocol feature

VRRP provides hosts with redundant routers in the network topology without any need for the hosts to reconfigure or know that there are multiple routers. If the primary (master) router fails, a secondary router assumes control and continues to use the virtual router IP (VRIP) address. VRRP Route Interface Tracking extends the capability of VRRP to allow tracking of specific route/ interface IP states within the router that can alter the priority level of a virtual router for a VRRP group.

Chapter 4. Data Center Features

4.1. Priority-Based Flow Control

Ordinarily, when flow control is enabled on a physical link, it applies to all traffic on the link. When congestion occurs, the hardware sends pause frames that temporarily suspend traffic flow.

Pausing traffic helps prevent buffer overflow and dropped frames.

Priority-based flow control (PFC) provides a way to distinguish which traffic on physical link is paused when congestion occurs, based on the priority of the traffic. An interface can be configured to pause only high priority (i.e., loss-sensitive) traffic when necessary prevent dropped frames while allowing traffic that has greater loss tolerance to continue to flow on the interface.

Priorities are differentiated by the priority field of the IEEE 802.1Q VLAN header, which identifies an IEEE 802.1p priority value. In ICOS, these priority values must be mapped to internal class-of-service (CoS) values. To enable priority-based flow control for a particular CoS value on an interface:

1. Ensure that VLAN tagging is enabled on the interface so that the 802.1p priority values are carried through the network.

2. Ensure that 802.1p priority values are mapped to ICOS CoS values.

When priority-flow-control is disabled, the interface defaults to the IEEE 802.3x flow control setting for the interface. When priority-based flow control is enabled, the interface will not pause any CoS unless there is, at least, one no-drop priority.

4.2. OF-DPA OpenFlow Hybrid Switch Functionality Guide

4.2.1. Overview

An OpenFlow hybrid switch supports both OpenFlow operation and "normal" Ethernet switching operation. Various models for organizing OpenFlow hybrid switches are possible. The OF-DPA OpenFlow Hybrid Switch is one specific implementation of an OpenFlow hybrid switch. Other OpenFlow hybrid switch designs are possible. Such alternate designs may use different operational and configuration methods than the OF-DPA OpenFlow Hybrid Switch model.

The model employed in the OF-DPA OpenFlow Hybrid Switch is referred to as "Ships in the Night". In this model, the physical switch is partitioned by assigning ports to either the OpenFlow switch or the traditional switch (bridge_normal). This is accomplished using an OVSDB Bridge table entry representing the OF-DPA OpenFlow pipeline. This bridge entry has the datapath_type column set to "ofdpa". The OpenFlow related code in the switch driver plugin uses the datapath_type value to verify that OpenFlow configuration is installed on ports assigned to the OF-DPA bridge.

According to the OpenFlow Switch Specification, an OpenFlow hybrid switch should provide a classification mechanism that routes traffic to either the OpenFlow pipeline or the normal pipeline. The mechanism used in the OF-DPA OpenFlow Hybrid Switch is based on the port the packet enters the switch. A port is under the control of the OF-DPA pipeline when the port is associated with an entry in the bridge table whose datapath_type is "ofdpa".

Once configured, the switch contains two independent forwarding pipelines. One pipeline is an OpenFlow pipeline that processes packets based on OpenFlow policy contained in the flow and group table entries installed by an OpenFlow Controller. The other pipeline is the traditional OPS forwarding pipeline. Packets entering physical ports are operated on by the forwarding pipeline corresponding to the bridge configured to contain the port. Following the "Ships in the Night" model, packets remain within a single pipeline and egress a physical port also assigned to the same pipeline.

In order to support the OF-DPA OpenFlow Hybrid Switch, the ASIC plugin implements the APIs required to support the OF-DPA pipeline model. One ASIC plugin that supports this is the OpenNSL plugin. Please see DESIGN.md in the ops-switchd-opennsl-plugin repository for more information. Other plugins may also support the OF-DPA OpenFlow Hybrid Switch feature in later releases.

4.2.2. OpenFlow Pipeline

An OpenFlow switch’s pipeline is described by its Table Type Pattern (TTP) as defined by the Open Networking Foundation (ONF). A TTP is an abstract switch model that describes specific switch forwarding behaviors that an OpenFlow controller can program via the OpenFlow-Switch protocol. A TTP represents the flow processing capabilities of an OpenFlow Switch.

The OF-DPA pipeline supports the configuration of L2 bridges programmed by OpenFlow controllers. The L2 bridges may be programmed to isolate virtual tenant networks on shared network infrastructure.

Packets are assigned to a virtual tenant by classifying packets based on port, or the combination of port and VLAN. This assignment is done by programming the VLAN table to set the Tunnel-ID metadata for the packet. This Tunnel-ID is used in place of a VLAN ID to look up the forwarding destination. The controller programs a Bridging table flow entry to match the MAC address and the Tunnel-ID.

The flow tables used by the OF-DPA pipeline are shown in the following diagram.

```
            +--------+     +--------+    +-------------+
            |        |     |        |    |             |
+------+    | Ingress|     | VLAN   |    | Termination |
| port +----> Port   +----->        +----> MAC         +--+
+------+    |        |     |        |    |             |  |
            |        |     |        |    |             |  |
            |        |     |        |    |             |  |
            +--------+     +--------+    +-------------+  |
                                                          |
         +------------------------------------------------+
         |  +----------+    +-----------+
         |  |          |    |           |
         |  | Bridging |    | Policy    |    +---------+    +------+
         +-->          +----> ACL       +----> actions +----> port |
            |          |    |           |    +---------+    +------+
            |          |    |           |
            |          |    |           |
            +----------+    +-----------+
```

Not all of the tables in this diagram are active in this version. Some are placeholders for future use. The inactive flow tables have built-in default flow entries for now and cannot be programmed with flow entries. OpenFlow flow entries contain a "Goto-Table" instruction with specific table ID number. Including tables that will be used in future implementations helps preserve backward compatibility with OpenFlow configurations used in the current TTP.

The flow table IDs for each table are:

Table Name	Table ID
Ingress Port	0
VLAN	10
Termination MAC	20
Bridging	50
Policy ACL	60

4.2.3. Ingress Port Flow Table

This is a placeholder flow table. No flows can be added to this table by the controller.

4.2.4. VLAN Flow Table

Flows in this table match on port or port and VLAN. The port must be assigned to the OF-DPA bridge for the flow to be added. The set-field action setting the tunnel-id metadata is applied to matching packets. The Goto-Table instruction must specify the Termination MAC flow table.

name	IN_PORT	match_type	VLAN_VID	match_type	GOTO_TABLE	APPLY_ACTIONS
Tunnel Assignment - VLAN Tagged	<ofport>	exact	<vid> 0x1000	exact	Termination MAC	SET_FIELD TUNNEL_ID <tunnel_id>
Tunnel Assignment - Untagged	<ofport>	exact	0	exact	Termination MAC	SET_FIELD TUNNEL_ID <tunnel_id>
Tunnel Assignment - Priority Tagged	<ofport>	exact	0x1000	exact	Termination MAC	SET_FIELD TUNNEL_ID <tunnel_id>
Tunnel Assignment - All On Port	<ofport>	exact			Termination MAC	SET_FIELD TUNNEL_ID <tunnel_id>

4.2.5. Termination MAC Flow Table

This is a placeholder flow table. No flows can be added to this table by the controller.

4.2.6. Bridging Flow Table

Flows in this table match on tunnel-id and destination MAC. The flow entry must include a group action in the write-actions instruction. The Goto-Table instruction must specify the Policy ACL flow table.

name	ETH_DST	match_type	TUNNEL_ID	match_type	GOTO_TABLE	WRITE_ACTIONS
Unicast Overlay Bridging	<mac>	exact	<tunnel_id>	exact	Policy ACL	GROUP <L2 Interface>

4.2.7. Policy ACL Flow Table

This is a placeholder flow table. No flows can be added to this table by the controller.

4.2.8. L2 Interface Group Entry

The current TTP uses one type of group entry. This is called an L2 Interface group.

In OF-DPA, group ID is used to convey information about the group entry contents. Part of this information is the group entry type within OF-DPA. L2 Interface Group entries are assigned type == 0.

The group ID for this type of group entry is made up of the following fields:

bits:	31:28	27:16	15:0
content:	Type	VLAN ID	Port

As an example, the ID for an L2 Interface group entry that specifies VLAN ID 100 (0x64) and port 7 (0x0007) is 6553607 (0x00640007).

The action bucket for an L2 Interface Group entry specifies the port the packet is transmitted from. The port must be assigned to the OF-DPA bridge. The action set may also include the pop_vlan action which causes packets to be sent untagged.

4.2.9. OF-DPA OpenFlow Hybrid Switch (OF-DPA) Configuration

The OF-DPA OpenFlow Hybrid Switch feature is configured by adding and updating entries in the OVSDB. The Bridge, Port, Interface, and Controller tables are used. There are multiple ways to change OVSDB content. These include the ovs-vsctl utility, the Swagger UI RESTful API, and from a remote system using the OVSDB protocol (RFC 7047).

The following sections show the steps to configure the switch for OpenFlow Hybrid Switch operation. The examples shown use invocations of ovs-vsctl from the OPS switch's console. More information about ovs-vsctl commands is available in the man pages for this utility. These configuration examples illustrate the OVSDB content required and can be used to determine how the same elements are configured using other methods.

4.2.10. Creating the OF-DPA Bridge Table Entry

The OF-DPA pipeline is represented by an entry in the Bridge table. OPS automatically creates and configures an entry in the Bridge table named bridge_normal. A second Bridge table entry is created for the OF-DPA pipeline. The name of this bridge is not important, but the entry's datapath_type must be set to ofdpa. In the following examples, the bridge representing the OF-DPA pipeline is named bridge_ofdpa.

The following example adds a bridge named bridge_ofdpa and sets its datpath_type to ofdpa.

```
root@switch:~# ovs-vsctl add-br bridge_ofdpa
root@switch:~# ovs-vsctl set Bridge bridge_ofdpa datapath_type=ofdpa
```

4.2.11. Creating Port Table Entries Assigned to the OF-DPA Bridge

As discussed above, the method used to determine which pipeline processes a packet is by the port the packet ingresses. In order for a packet to be handled by the OF-DPA pipeline, it must enter the switch via a port assigned to the OF-DPA bridge.

The following example assigns ports 1 and 2 to the OF-DPA bridge called bridge_ofdpa.

```
root@switch:~# ovs-vsctl add-port bridge_ofdpa 1
root@switch:~# ovs-vsctl add-port bridge_ofdpa 2
```

The ports are associated with entries in the Interface table. The Interface table entry with the same name as the port is bound to the port. After the port is added to the bridge, the interface is assigned an OpenFlow port number. The OpenFlow port number value, used in all OpenFlow configuration to identify the port, is recorded in the ofport column of the Interface table entry.

In the example above, the resulting OVSDB content is:

```
root@switch:~# ovsdb-client dump Interface _uuid name ofport
Interface table
_uuid                                name          ofport
------------------------------------ ------------- ------
7553f083-eace-4ea0-8618-e3abb39d4764 "1"           1
6b5fc1a1-0b8a-49dd-95f0-21aa8b9b8b56 "10"          []
c5d786b8-c41a-4e0b-b52c-b6342e39a297 "11"          []
db9ea0c3-abab-4162-9aab-d4ebd60b0ee8 "12"          []
7f2509f0-dee5-4d32-8f70-4cc5b6d4290f "13"          []
e194c580-de9a-453a-ab35-0a348a861965 "14"          []
6d4d8cf7-3b8d-485f-959c-e956fd699e10 "15"          []
1462895e-6e2d-4b76-82d5-0895c60b29fe "16"          []
aafb2a54-2bf8-4eb9-993b-fb8973d5a427 "17"          []
7dc31b97-66dc-4b13-9994-267c7d1ab663 "18"          []
b8feb9aa-b65f-4cb0-bc81-79b8d87adb9b "19"          []
7b7cace3-4ce2-4404-8122-0b0471e640df "2"           2
...
```

The initial implementation assigns OpenFlow port numbers based on the order the ports are added to the bridge. In the case above, port 1 is first port to be added to the OpenFlow Bridge so it is assigned ofport == 1. The same follows for port 2 getting ofport == 2. However, since the ofport value assignment is automatically generated, the ofport value may not correspond to the port name. That is, continuing the example, if port 14 is added to the OpenFlow bridge next, it will be assigned ofport == 3. This creates a dependency between the configuration order of the OVSDB content and the ofport values assigned. This is acceptable for experimentation with the initial OpenFlow Hybrid Switch feature, but an enhancement is planned to make the ofport assignment deterministic. After that change, the ofport value will be the same as the port name (that is, the Interface table entry associated with the Port table entry named "14" will be assigned ofport == 14).

4.2.12. Enabling Interfaces Associated with Ports

Interfaces associated with ports added to the OF-DPA bridge need to be administratively enabled to transition to link up. This is done by setting the admin key in the user_config column of the Interface table entries.

For example:

```
root@switch:~# ovs-vsctl add Interface 1 user_config admin=up
root@switch:~# ovs-vsctl add Interface 2 user_config admin=up
```

4.2.13. Configuring the OpenFlow Agent to Communicate with OpenFlow Controllers

Communication with an OpenFlow controller is configured by updating the Controller table. OpenFlow messages are sent to the switch to configure the OpenFlow pipeline. There are many options regarding how the agent and controller communicate. For experimentation, it is possible to use the ovs-ofctl utility on the switch’s console to send OpenFlow messages to the agent on the same switch. However, in most cases, the OpenFlow controller is another system.

Example configuring the agent to accept an OpenFlow connection from a controller with a specific IP address and the default port over TCP:

```
root@switch:~# ovs-vsctl set-controller bridge_ofdpa
tcp:[OpenFlow Controller IP]
```

Example configuring the agent to accept an OpenFlow connection from a controller at any IP address and the default port over TCP:

```
root@switch:~# ovs-vsctl set-controller bridge_ofdpa ptcp:
```

4.2.14. Using the OpenFlow Protocol to Configure Forwarding

Once the OpenFlow Hybrid switch is configured, various OpenFlow controllers can be used to program the OpenFlow pipeline. These include ODL, Ryu, dpctl, ovs-ofctl, and many others. Examples here use the ovs-ofctl utility on the local switch.

The following example sets up the OF-DPA pipeline to forward packets matching the flow configuration to be switched from port 1 to port 2. The processing steps in the OF-DPA pipeline are:

1. Assign packets tagged with VLAN 100 arriving on port 1 to to tunnel_id 343
2. Packets assigned to tunnel_id 343 with a destination MAC equal to 00:00:00:00:00:11 are handled using L2 Interface entry 0x00640002
3. The L2 Interface entry specifies the packets are sent out port 2 tagged

   ```
   root@switch:~# ovs-ofctl -O OpenFlow13 add-group bridge_ofdpa
   group_id=0x00640002,type=all,bucket=output:2
   root@switch:~# ovs-ofctl -O OpenFlow13 add-flow bridge_ofdpa
   table=10,in_port=1,dl_vlan=100,actions=set_field:343-\>tunnel_id,
   goto_table:20
   root@switch:~# ovs-ofctl -O OpenFlow13 add-flow bridge_ofdpa table=50,
   tunnel_id=343,dl_dst=00:00:00:00:00:11,actions=group:0x00640002,goto_table:60
   ```

4.2.15. Configuration Example

This section shows a simple use case and how it is configured using OpenFlow. The diagram below illustrates the network used for this example.

```
+----------+                                                    +----------+
|          |                                                    |          |
| Server A +--------+                                    +------+ Server C |
|          |        |1 +----------+        +----------+ 1|      |          |
+----------+        +--+          |        |          +--+      +----------+
MAC:                   |          | 3    3 |          |                 MAC:
00:00:00:00:00:11      | Switch 1 +--------+ Switch 2 |    00:00:00:00:00:33
                     2 |          |        |          | 2
```

```
+----------+        +--+          |        |          +--+      +----------+
|          |        |  +----------+        +----------+  |      |          |
| Server B +--------+                                    +------+ Server D |
|          |                                                    |          |
+----------+                                                    +----------+
MAC:                                                                    MAC:
00:00:00:00:00:22                                          00:00:00:00:00:44
```

In this use case, packets from the servers are sent tagged. Servers A and C are able to communicate and servers B and D are able to communicate. On the link between switches 1 and 2, packets are sent with VLAN tags. Packets sent between server A and server C are sent on this link using VLAN 100 and packets between server B and D are sent using VLAN 200. This results in the traffic between the one pair of servers being isolated from the other pair.

Following a packet through the OF-DPA pipeline from server A to server C the packet is presented to each flow table in order.

First, in switch 1, the VLAN flow table matches on the packet's incoming ofport and whether it arrived tagged. In this case the flow matches packets entering ofport == 1 that are tagged with VLAN ID 100. The action for the matching flow entry is to perform a SET_FIELD TUNNEL_ID action setting the packet's TUNNEL_ID metadata to 343. The goto instruction links to the Termination MAC flow table which is inactive in this version. The Termination MAC table has a default action of goto Bridging.

At the Bridging flow table, the flow entries match on the packet's destination Ethernet MAC address and the TUNNEL_ID metadata assigned in the VLAN flow table. Packets from server A to server C have a destination MAC address of 00:00:00:00:00:33 and TUNNEL_ID of 343. Matching packets are handled by the L2 Interface group entry in the flow. In this case the L2 Interface group entry's ID is 0x00640003.

The L2 Interface group entry (0x00640003) specifies that packets using the group entry are sent out port 3 tagged with the VLAN tag equal to 100.

On switch 2, the VLAN flow table matches the packet arriving on port 3 with a VLAN tag equal to 100. The actions for this flow entry are to SET_FIELD TUNNEL_ID to 343. In the Bridging table, packets with destination MAC of 00:00:00:00:00:33 and TUNNEL_ID of 343 are handled by the L2 Interface group entry 0x00640001.

The L2 Interface group entry (0x00640001) specifies that packets using the group entry are sent out port 1 with a VLAN tag.

The other paths between servers follow a similar path with the appropriate TUNNEL_ID and VLAN ID values used.

The following shows the commands to configure the OF-DPA pipeline using the ovs-vsctl and ovs-ofctl utilities.

On both switches:

```
root@switch:~# ovs-vsctl add-br bridge_ofdpa
root@switch:~# ovs-vsctl set Bridge bridge_ofdpa datapath_type=ofdpa
root@switch:~# ovs-vsctl add-port bridge_ofdpa 1
root@switch:~# ovs-vsctl add-port bridge_ofdpa 2
root@switch:~# ovs-vsctl add-port bridge_ofdpa 3
```

```
root@switch:~# ovs-vsctl add Interface 1 user_config admin=up
root@switch:~# ovs-vsctl add Interface 2 user_config admin=up
root@switch:~# ovs-vsctl add Interface 3 user_config admin=up
```

```
 root@switch:~# ovs-ofctl -O OpenFlow13 add-group bridge_ofdpa
 group_id=0x00640001,type=all,bucket=output:1
root@switch:~# ovs-ofctl -O OpenFlow13 add-group bridge_ofdpa
group_id=0x00c80002,type=all,bucket=output:2
root@switch:~# ovs-ofctl -O OpenFlow13 add-group bridge_ofdpa
group_id=0x00640003,type=all,bucket=output:3
root@switch:~# ovs-ofctl -O OpenFlow13 add-group bridge_ofdpa
group_id=0x00c80003,type=all,bucket=output:3
```

```
root@switch:~# ovs-ofctl -O OpenFlow13 add-flow bridge_ofdpa table=10,
in_port=1,dl_vlan=100,actions=set_field:343-\>tunnel_id,goto_table:20
root@switch:~# ovs-ofctl -O OpenFlow13 add-flow bridge_ofdpa table=10,
in_port=3,dl_vlan=100,actions=set_field:343-\>tunnel_id,goto_table:20
```

```
root@switch:~# ovs-ofctl -O OpenFlow13 add-flow bridge_ofdpa table=10,
in_port=2,dl_vlan=200,actions=set_field:632-\>tunnel_id,goto_table:20
root@switch:~# ovs-ofctl -O OpenFlow13 add-flow bridge_ofdpa table=10,
in_port=3,dl_vlan=200,actions=set_field:632-\>tunnel_id,goto_table:20
```

On Switch 1:

```
root@switch:~# ovs-ofctl -O OpenFlow13 add-flow bridge_ofdpa table=50,
tunnel_id=343,dl_dst=00:00:00:00:00:11,actions=group:0x00640001,goto_table:60
root@switch:~# ovs-ofctl -O OpenFlow13 add-flow bridge_ofdpa table=50,
tunnel_id=343,dl_dst=00:00:00:00:00:33,actions=group:0x00640003,goto_table:60
root@switch:~# ovs-ofctl -O OpenFlow13 add-flow bridge_ofdpa table=50,
tunnel_id=632,dl_dst=00:00:00:00:00:22,actions=group:0x00c80002,goto_table:60
root@switch:~# ovs-ofctl -O OpenFlow13 add-flow bridge_ofdpa table=50,
tunnel_id=632,dl_dst=00:00:00:00:00:44,actions=group:0x00c80003,goto_table:60
```

On Switch 2:

```
root@switch:~# ovs-ofctl -O OpenFlow13 add-flow bridge_ofdpa table=50,
tunnel_id=343,dl_dst=00:00:00:00:00:33,actions=group:0x00640001,goto_table:60
root@switch:~# ovs-ofctl -O OpenFlow13 add-flow bridge_ofdpa table=50,
tunnel_id=343,dl_dst=00:00:00:00:00:11,actions=group:0x00640003,goto_table:60
root@switch:~# ovs-ofctl -O OpenFlow13 add-flow bridge_ofdpa table=50,
tunnel_id=632,dl_dst=00:00:00:00:00:44,actions=group:0x00c80002,goto_table:60
root@switch:~# ovs-ofctl -O OpenFlow13 add-flow bridge_ofdpa table=50,
tunnel_id=632,dl_dst=00:00:00:00:00:22,actions=group:0x00c80003,goto_table:60
```

4.3. VXLAN

VXLAN is one method of creating tenant networks on a common network infrastructure. VXLAN encapsulates Ethernet frames in IP packets, thus enabling the network to provide the illusion that hosts connected to arbitrary access routers are attached to a common layer-2 networks. The VXLAN encapsulation includes a 24-bit virtual network ID (VNID). Hosts can be associated to a VNID and restricted to communicate only with hosts associated to the same VNID. This association segregates communities of interest, or tenants, into different virtual networks. VXLAN allows a public or private data center operator to use a common network infrastructure to provide virtual private network service to multiple tenants while distributing any given tenant’s compute and storage resources anywhere in the network infrastructure.

In a data center, VXLAN encapsulation and decapsulation of tenant packets is normally done by a virtual switch within a virtualized server; however, not all tenant systems are virtualized. Non-virtualized tenant systems can participate in a VXLAN by using a VXLAN gateway. A VXLAN gateway is a networking device that does VXLAN encapsulation and decapsulation. A server’s first-hop router, often referred to as a top-of-rack (ToR) device, can be a VXLAN gateway.

With VXLAN, the inner Ethernet header can optionally include an incoming VLAN tag. The DCVPN application always strips the inner VLAN information from the incoming Ethernet packet during encapsulation. The inner payload in the VXLAN encapsulated packet does not contain the incoming VLAN tag information in it, which enables flexibility in mapping available VLANs to VNIDs.

The allowed range of VNID values is 1–16777214. VNID 16777215 is reserved for internal purposes.

Chapter 5. Quality of Service Features

5.1. Overview

Quality of Service (QoS) features allow network devices the ability to customize servicing behaviors to different kinds of traffic reflecting each traffic type's unique characteristics and importance your organization:

- Ensure uniform and efficient traffic-handling throughout the network, while keeping the most important traffic moving at an acceptable speed, regardless of current bandwidth usage.
- Exercise control over the priority settings of inbound traffic arriving at each network device.

Packets traverse a network device in 4 stages. QoS configuration affects stages 1, 3, and 4:

1. Initial prioritization
   a. QoS trust mode
2. Destination determination
3. Queuing
   a. QoS queue profile
4. Transmission Scheduling
   a. QoS schedule profile

Besides the packet itself, network devices keep other information regarding the packet collectively called *metadata*:

- arrival port
- arrival VLAN and/or VRF
- destination port
- destination VLAN and/or VRF
- local-priority
- color
- etc

QoS functionality within the network device uses local-priority and color metadata:

Local-priority is generally one of 8 levels (0-7). Zero is the lowest priority. The allowed maximum will vary per product family. It is used to determine which queues a packet will use.

Color is one of three values (0-2), commonly named green (0), yellow (1), and red (2). These are mostly used with packets marked with Assured Forwarding (AF) DSCP values. The default is green (0).

In summary:

- QoS trust mode configures how arriving packets are assigned the initial values of their local-priority and color.
- QoS queue profiles configures which queues packets will use awaiting transmission.
- QoS schedule profiles configures the order of queues selected to transmit packets.

5.1.1. End-to-end behavior

The QoS configuration of each network device along the path between the source and destination must be aligned to achieve the desired end-to-end behavior. There are three basic service schemes for end-to-end demarcating different types of traffic:

- Best Effort Service
- Ethernet Class of Service (CoS)
- Internet Differentiated Services (DiffServ)

These three are not mutually exclusive. Different ports can use different service behaviors. For your network as a whole, it is best to select one to use as the primary end-to-end behavior. The other two should only be used on an exception basis.

5.1.1.1. Best effort service

This is the simplest behavior. All traffic is treated equally in a first-come, first-served manner. If the traffic load is low in relation to the capacity of the network links, then there is no need for the administrative complexity and costs of maintaining an end-to-end policy. This is sometimes called *over provisioning* - all link speeds are well higher than the peak loads.

5.1.1.2. Ethernet class of service

The Ethernet standard 802.1Q provides a means to mark packets with one of eight Classes of Service (CoS). A 3-bit Priority Code Point field is within the 16-bit Ethernet VLAN tag to mark the packet’s class of service:

```
+--------+--------+--------+----------+-----------+--------
| mac-da | mac-sa | 0x8100 | VLAN tag | ethertype | data...
+--------+--------+--------+----------+-----------+--------
                         /            \
                        /              \
                       /                \
                    +-----+-----+---------+
                    | pcp | dei | vlan_id |
                    +-----+-----+---------+
```

The standard recommends the types of traffic that should use each of the eight classes of service based on their behavior aggregate:

CoS	Traffic Type	Example Protocols
7	Network Control	STP, PVST
6	Internetwork Control	BGP, OSPF, PIM

CoS	Traffic Type	Example Protocols
5	Voice (<10ms latency)	VoIP(UDP)
4	Video (<100ms latency)	RTP
3	Critical Applications	SQL RPC, SNMP
2	Excellent Effort	NFS, SMB
0	Best Effort	HTTP, TELNET
1	Background	SMTP, IMAP

Notice that CoS 1 is the lowest CoS and zero is the next higher. This was deliberate to allow specifying traffic below default (Best Effort) traffic.

5.1.1.3. Internet differentiated services

This is the most sophisticated behavior. Internet Protocol standards (RFC) provides a means to mark packets with one of sixty four service classes, via the Differentiated Services Code Point (DSCP). The DSCP value is carried within the upper 6-bits of 8-bit IPv4 Type-of-Service (ToS) or IPv6 Traffic Class (TC) header fields.

```
IPv4
+-----+-----+-----+----+--------+-----+-------+--------+-------|-------+----
| ver | tos | len | id | offset | ttl | proto | chksum | ip-sa | ip-da |data
+-----+-----+-----+----+--------+-----+-------+--------+-------|-------+----
     /       \
    /         \
   +-----+-----+
   |dscp | ecn |
   +-----+-----+
    \         /
 IPv6\       /
+-----+-----+-----+-------+-------------+-----------+-------+-------+-----
| ver | tc  | len | label | next_header | hop_limit | ip-sa | ip-da | data
+-----+-----+-----+-------+-------------+-----------+-------+-------+-----
```

More standards specify the per-hop behavior (PHB) for many of the code points that is beyond the scope of this document. The Wikipedia Differentiated Services article is a good starting place.

DSCP	Name	Service Class	RFC
56	CS6	Network Control	2474
46	EF	Telephony	3246
40	CS5	Signaling	2474
34,36,38	AF41,AF42,AF43	Multimedia Conferencing	2597
32	CS4	Real-Time Interactive	2474
26,28,30	AF31,AF32,AF33	Multimedia Streaming	2597
24	CS3	Broadcast Video	2474
18,20,22	AF21,AF22,AF23	Low-Latency Data	2597

DSCP	Name	Service Class	RFC
16	CS2	OAM	2474
10,12,14	AF11,AF12,AF13	High-Throughput Data	2597
00	CS0,BE,DF	Standard	2474
08	CS1	Low-Priority Data	3662

Notice that DSCP CS1 (8) is the lowest priority and CS0 (0) is the next higher. This was deliberate to allow specifying traffic below standard (best-effort or default-forwarding).

5.1.2. Queuing and scheduling

When using end-to-end behavior Ethernet Class of Service or Internet Differentiated Services, different priorities of traffic must be placed in different queues so the network device can service them appropriately. Separate queues allow delay or jitter sensitive traffic to be serviced before more less-time critical or bulk traffic.

Queue profiles

Queue policies configure the queues different priorities of traffic will use. Queues are numbered in priority order with zero being the lowest priority. The larger the queue number the higher priority of the queue.

Schedule profiles

Schedule policies configure the order of the queues that packets are taken off (de-queued) to be transmitted. The schedule discipline is the algorithm the scheduler employs each time (round) it must select the next packet for transmission.

Strict priority (SP)

This is the simplest of scheduling disciplines. For each round, the scheduler will select the packet from the highest priority (numbered) queue for transmission. Effectively, it will always empty all packets from the highest priority queue before any other lower priority queue.

While this does provide prioritization of traffic, when spikes of high priority traffic occur it will prevent lower priority traffic from being transmitted (aka *queue starvation*).

Deficit weighted round robin (DWRR)

Deficit weighted round robin can limit queue starvation by providing a fairer distribution of available bandwidth across the priorities. Lower priority queues will have some service even when packets are present in higher priority queues. The degree to which this occurs depends on the weights assigned to each queue.

Each non-empty queue has a deficit counter, which is used to track the amount of bytes it is allowed to send. The counter starts at zero when a packet arrives at an empty queue.

At the start of the round, the deficit counters of all non-empty queues are incremented by a quantum value in proportion to their weight. Then the scheduler will inspect the non-empty queues in decreasing priority order, comparing the queue’s deficit counter against the packet size at the head of the queue. The round ends when a deficit counter is found that is larger than the packet size. That packet is de-queued for transmission and the queue’s deficit counter is decremented.

In general, the deficit counters of lower priority queues will be larger (i.e. they build up a deficit) than the deficit counters of higher priority queues. With a mix of large and small packet sizes of high priority traffic, the deficit counters of higher priority queues will have some rounds where they will be smaller than the size of the queue at head of queue. This occasionally allows traffic from lower priority queues to be selected for transmission.

Strict priority plus deficit weighted round robin

SP plus DWRR is a hybrid of two disciplines primarily used in networks carrying voice traffic. Queuing delay and jitter (i.e. variance in delay) can readily affect the quality of a voice call. To prevent these conditions, voice traffic should be in the highest priority queue, but what of the traffic in other queues?

Using strict priority for all queues would be good for the voice, but increases the risk of lower priority queue starvation. DWRR for all queues would be fairer to the lower priority traffic, but at the risk of increased voice traffic delay and jitter.

SP plus DWRR solves both problems. The highest priority queue is scheduled using strict priority, while the remaining seven lower priority queues use DWRR discipline.

On each round, the scheduler first checks the highest priority queue. When it has a packet ready, that packet will selected for immediate transmission. When the highest priority queue is empty, the scheduler uses DWRR discipline to fairly select the next packet for transmission from the remaining lower priority queues.

5.1.3. Definition of terms

Term	Description
Class	For networking, a set of packets sharing some common characteristic (e.g. all IPv4 packets)
Codepoint	Used in two different ways — either as the name of a packet header field or as the name of the values carried within a packet header field: Example 1: Priority code point (PCP) is the name of a field in the IEEE 802.1Q VLAN tag. Example 2: Differentiated services codepoint (DSCP) is the name of values carried within the DS field of the header field.
Color	A metadata label associated with each packet within the switch with three values: "green", "yellow", or "red". It is used by the switch when packets encounter congestion for resource (queue) to distinguish which packets should be dropped. It is used by the switch when packets encounter congestion for resource (queue) to distinguish which packets should be dropped.
Class of service (CoS)	A 3-bit value used to mark packets with one of eight classes (levels of priority). It is carried within the priority code point (PCP) field of the IEEE 802.1Q VLAN tag.
Differentiated services codepoint (DSCP)	A 6-bit value used to mark packets for different per-hop behavior as originally defined by IETF RFC 2474. It is carried within the differentiated services (DS) field of the IPv4 or IPv6 header.
Local-priority	A meta-data label associated with a packet within a network switch. It is used by the switch to distinguish packets for different treatment (e.g. queue assignment, etc.)

Term	Description
Metadata	Information labels associated with each packet in the switch, separate from the packet headers and data. These labels are used by the switch in its handling of the packet. Examples: arrival port, egress port, VLAN membership, local priority, color, etc.
Priority code point (PCP)	The name of a 3-bit field in the IEEE 802.1Q VLAN tag. It carries the CoS value to mark a packet with one of eight classes (priority levels).
Quality of service (QoS)	General term used when describing or measuring performance. For networking, it means how different classes of packets are treated across the network or device. For more information, see https://en.wikipedia.org/wiki/Quality_of_service.
Traffic class (TC)	General term for a set of packets sharing some common characteristic. It used to be the name of an 8-bit field in the IPv6 header originally defined by IETF RFC 2460. This field name was changed to differentiated services by IETF RFC 2474.
Type of service (ToS)	General term when there are different levels of treatment (e.g. fare class). It used to be the name of an 8-bit field in the IPv4 header originally defined by IETF RFC 791. This field name was changed to differentiated services by IETF RFC 2474

5.1.4. Configuring QoS trust

QoS trust mode configures the network device end-to-end behavior selected by your organization. The purpose is to set the initial value of the local-priority and color packet metadata.

The trust mode for all ports can be set in the top-level (global) configuration context with the command qos trust [none | cos | dscp]. The command no qos trust will revert the trust mode back to the initial trust mode. Use the command show qos trust default at any time to view the initial trust mode.

There must always be a trust mode configured for every port. Each product automatically provisions OpenSwitch with a default trust mode for all ports. Use the command **show qos trust** to view the current trust mode. Do not use show running-configuration as it will only display changes from the default values.

Each port can have its own trust mode configured. This will override the top-level (global) trust mode. In the interface configuration context, enter the command qos trust [none | cos | dscp]. To revert the port back to using the default trust mode, in the interface configuration context enter the command no qos trust.

The only way to remove qos trust mode from the running configuration display is to revert back to the initial trust mode via no qos trust. The same is true for port overrides: Only no qos trust will remove the override from the running configuration display.

5.1.4.1. Configuring Ethernet class of service

To configure all ports to use Ethernet class of service:

```
#configure terminal
```

```
(config)#qos trust cos
```

To configure only one Ethernet port or all members of a link aggregation group (LAG):

```
#configure terminal
(config)# interface 1
(config-if)# qos trust cos
(config)# interface lag 100
(config-if)# qos trust cos
(config)# interface 2
(config-if)# lag 10
(config)# interface 3
(config-if)# lag 10
```

If qos trust cos is configured, the network device uses the CoS Map to determine which local-priority and color to assign the packet. To display the configuration of the CoS Map:

```
# show qos cos-map
code_point local_priority color   name
---------- -------------- ------- ----
0          1              green   Best_Effort
1          0              green   Background
2          2              green   Excellent_Effort
3          3              green   Critical_Applications
4          4              green   Video
5          5              green   Voice
6          6              green   Internetwork_Control
7          7              green   Network_Control
```

The above configuration follows IEEE 802.1Q standard assignments.

5.1.4.2. Configuring Ethernet 802.1D class of service

IEEE 802.1Q is the most current Ethernet standard for class of service. It superseded an earlier standard, 802.1D, in 2005. IEEE 802.1Q slightly changed the ordering of the classes of service from its predecessor IEEE 802.1D for CoS 2 and CoS 0:

CoS 802.1Q	CoS 802.1D
7 Network Control	7 Network Control
6 Internetwork Control	6 Voice (<10ms latency)
5 Voice (<10ms latency)	5 Video (<100ms latency)
4 Video (<100ms latency)	4 Controlled Load
3 Critical Applications	3 Excellent Effort
2 Excellent Effort	0 Best Effort
0 Best Effort	2 Spare
1 Background	1 Background

Note that in 802.1D, both CoS 2 and CoS 1 are below CoS 0 (Best Effort).

When an OpenSwitch device is installed in a network of devices following 802.1D class of service, the QoS Cos Map must be re-configured to follow the 802.1D standard by swapping the assignments of CoS 0 and 2:

```
#configure terminal
(config)#qos cos-map 0 local-priority 2 color green name Best_Effort
(config)#qos cos-map 2 local-priority 1 color green name Spare
# show qos cos-map
code_point local_priority color   name
---------- -------------- ------- ----
0          2              green   Best_Effort
1          0              green   Background
2          1              green   Spare
3          3              green   Critical_Applications
4          4              green   Video
5          5              green   Voice
6          6              green   Internetwork_Control
7          7              green   Network_Control
```

5.1.4.3. Configuring Internet DiffServ

To configure all ports to use Internet differentiated services:

```
#configure terminal
(config)#qos trust dscp
```

To configure only one Ethernet port or all the members of a link aggregation group (LAG):

```
#configure terminal
 (config)# interface 1
(config-if)# qos trust dscp
(config)# interface lag 100
(config-if)# qos trust dscp
(config)# interface 2
(config-if)# lag 10
(config)# interface 3
(config-if)# lag 10
```

For qos trust dscp, the network device uses the DSCP Map to determine which local-priority and color to assign the packet. Use show qos dscp-map to display the configuration of the DSCP Map.

5.1.5. Configuring queue profiles

The queue profile determines the assignment of local-priority to queues. A queue-profile must be configured on every port at all times.

OpenSwitch automatically provisions each network device with an initial queue profile named default. Use the command show qos queue-profile default to view the product default queue profile. Do not use show running-configuration as it will only display changes from the initial values.

The default queue-profile assigns each local-priority to the queue of the same number. This should work most all situations. A new queue profile can be created anytime by entering the command qos queue-profile NAME. Use map queue QUEUENUM local-priority PRI commands to assign local-priorities to queues.

Finally, use the apply qos queue-profile NAME schedule-profile NAME command to configure all ports to use the new profile.

5.1.6. Configuring schedule profiles

The schedule profile determines the order of queues selected to transmit a packet. A schedule profile must be configured on every port at all times.

OpenSwitch automatically provisions each network device with an initial schedule profile named default. Use the command show schedule-profile default to view the product default schedule profile. Do not use show running-configuration as it will only display changes from the initial values.

5.1.7. Configuring expedited forwarding using priority queuing

In many organizations, the network carries voice over IP (VoIP) traffic. It is delay and jitter sensitive, so dwell time in network devices must be kept to a minimum. It is critical that all network devices in the path have their per-hop behaviors configured identically to handle this traffic.

The objective is to configure the network device to have a dedicated queue just for voice traffic, and have that queue serviced before any other traffic.

As a prerequisite, voice traffic packets must be uniquely identified. Many networks using DiffServ mark voice packets with Expedited Forwarding (EF) DSCP.

To configure all ports to have DSCP EF packets, use the highest priority queue that is serviced first before any other queues' packets. Follow these five steps:

1. Change the default DSCP Map code point to local-priority assignments so EF has its own local-priority (5).
2. Configure a queue profile that puts the EF packets in the highest priority queue.
3. Configure an SP plus DWRR schedule profile.
4. Apply the queue and schedule profiles.
5. Configure global trust mode to DSCP.

Step 1: **Change the default DSCP Map for code points using local-priority 5**

The default DSCP Map has DSCP EF assigned to local-priority 5. It is necessary to have packets with DSCP EF be the sole user of local priority 5. The default DSCP Map has 8 code points, 40 through 47 (CS5), mapping to local-priority 5. The other seven code points have to be reassigned to another local-priority, depending on which protocol(s) are using these code points in your network. This example assumes CS5 is used by Call Signaling protocols and will be assigned local-priority 6.

```
#configure terminal
```

```
(config)# qos dscp-map 40 local-priority 6 color green name CS5
(config)# qos dscp-map 41 local-priority 6 color green
(config)# qos dscp-map 42 local-priority 6 color green
(config)# qos dscp-map 43 local-priority 6 color green
(config)# qos dscp-map 44 local-priority 6 color green
(config)# qos dscp-map 45 local-priority 6 color green
(config)# qos dscp-map 47 local-priority 6 color green
```

Now DSCP EF is the only code point using local-priority 5.

Step 2: **Create queue profile**

Create a queue profile that maps local-priority 5 to queue 7.

```
#configure terminal
(config)# qos queue-profile ef_priority
(config-queue)# name queue 7 Voice_Priority_Queue
(config-queue)# map queue 7 local-priority 5
(config-queue)# map queue 6 local-priority 7
(config-queue)# map queue 5 local-priority 6
(config-queue)# map queue 4 local-priority 4
(config-queue)# map queue 3 local-priority 3
(config-queue)# map queue 2 local-priority 2
(config-queue)# map queue 1 local-priority 1
(config-queue)# map queue 0 local-priority 0
```

Step 3: **Create an SP plus DWRR schedule profile**

Create a schedule profile that services queue 7 using strict priority (SP) and the remaining queues using DWRR). This example will give all DWRR queues equal weight. The actual weight values will be different in your network.

```
#configure terminal
(config)# qos schedule-profile sp_dwrr
(config-schedule)# strict queue 7
(config-schedule)# dwrr queue 6 weight 1
(config-schedule)# dwrr queue 5 weight 1
(config-schedule)# dwrr queue 4 weight 1
(config-schedule)# dwrr queue 3 weight 1
(config-schedule)# dwrr queue 2 weight 1
(config-schedule)# dwrr queue 1 weight 1
(config-schedule)# dwrr queue 0 weight 1
```

Step 4: **Apply the policies to all ports**

Apply the queue and schedule profiles to all ports from the configuration context.

```
#configure terminal
(config)# apply qos queue-profile ef_priority schedule-profile sp_dwrr
```

Step 5 **Configure DSCP trust mode to all ports**

```
#configure terminal
(config)# qos trust dscp
```

5.1.8. Monitoring queue operation

The show interface IFACE queues command will display the number of packets and bytes the queue has transmitted. Also displayed is the number of packets that were not transmitted.

```
switch# show interface 1 queues
Interface 1 is  (Administratively down)
Admin state is down
State information: admin_down
        Tx Packets             Tx Bytes  Tx Packet Errors
Q0             100                 8000                 0
Q1         1234567          12345678908                 5
Q2              0                     0                 0
Q3              0                     0                 0
Q4              0                     0                 0
Q5              0                     0                 0
Q6              0                     0                 0
Q7              0                     0                 0
```